FIG. 1

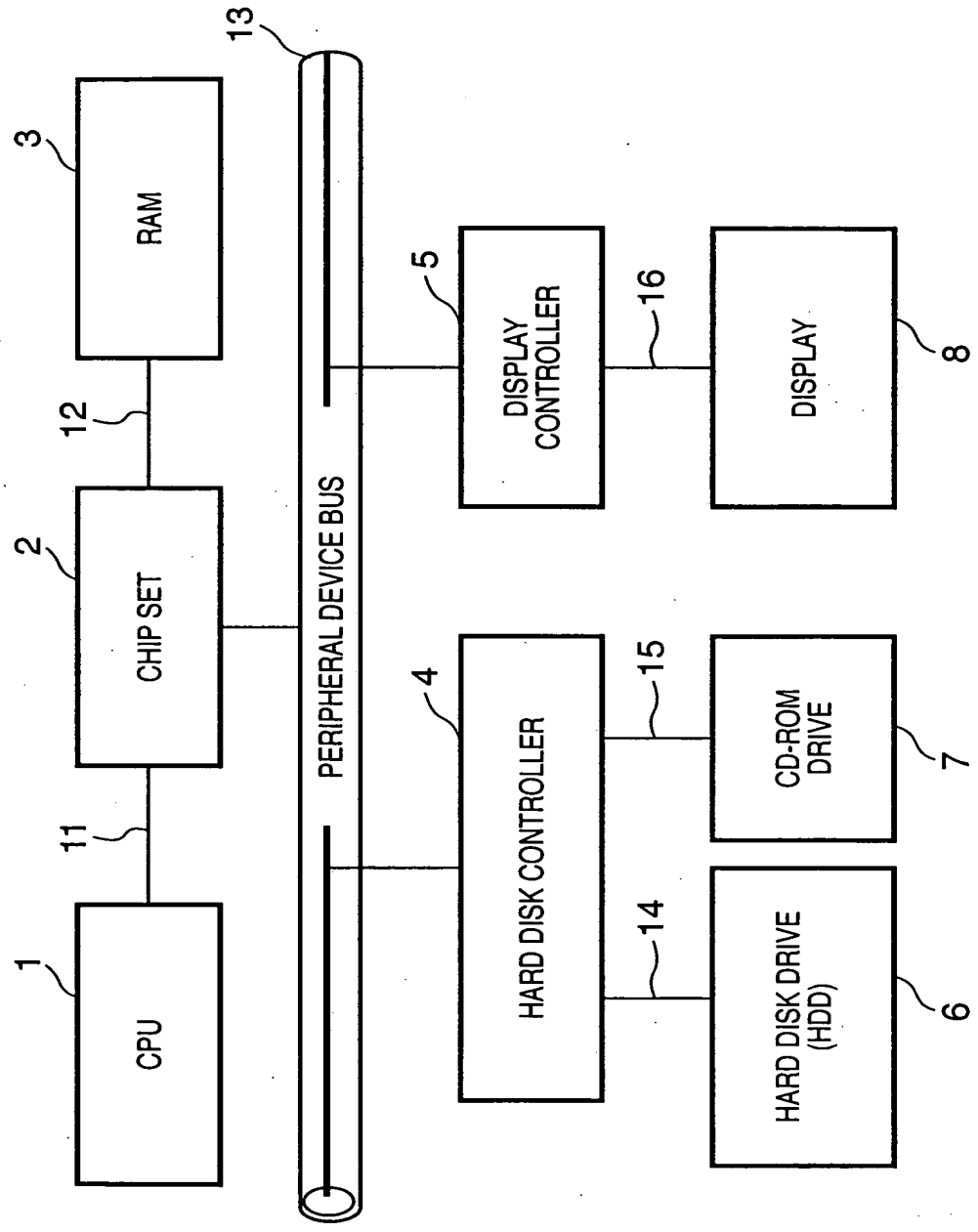

CPU
1
11
CHIP SET
2
12
RAM
3
13
PERIPHERAL DEVICE BUS
HARD DISK CONTROLLER
4
14
HARD DISK DRIVE (HDD)
6
15
CD-ROM DRIVE
7
DISPLAY CONTROLLER
5
16
DISPLAY
8

FIG. 2

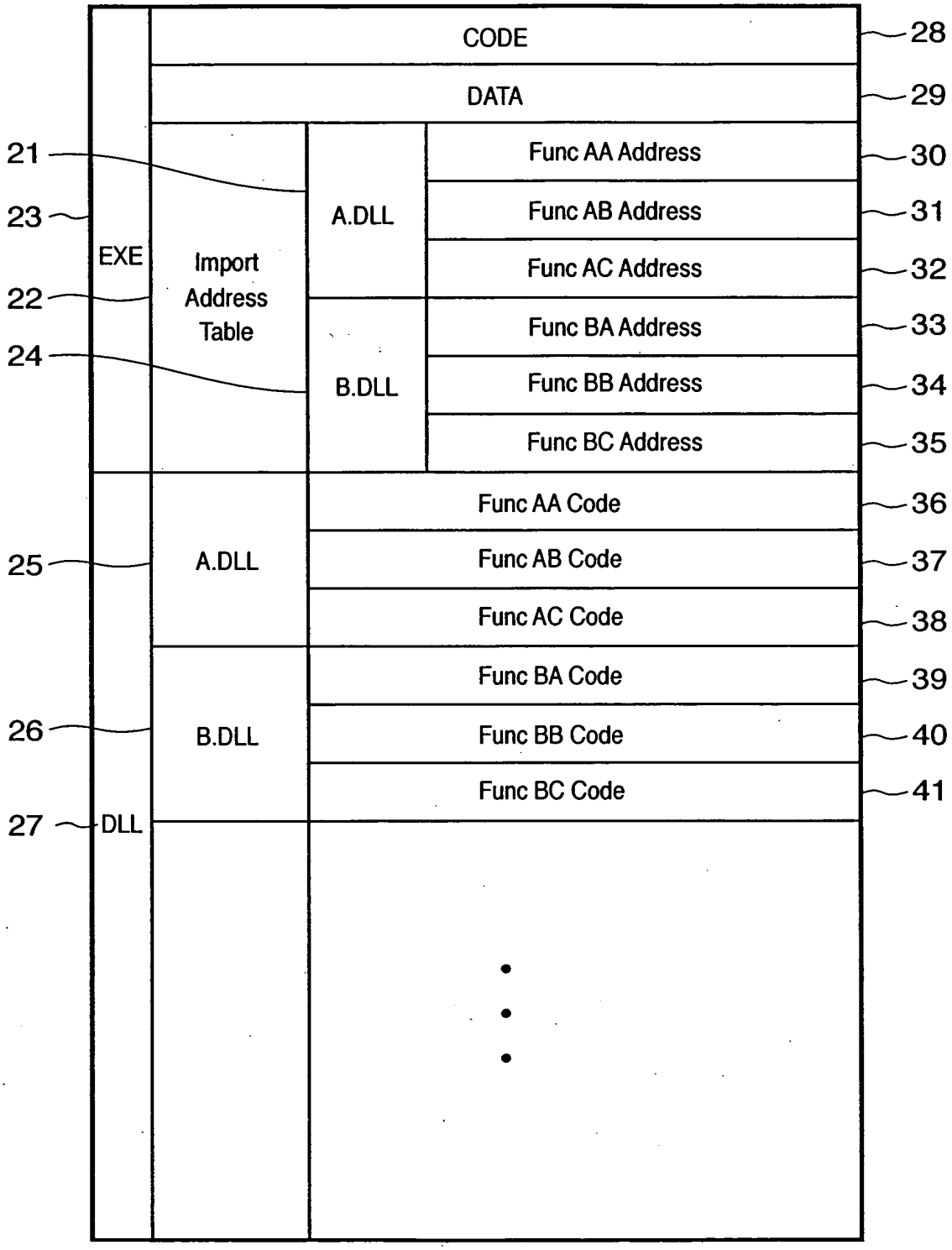

CODE
28
DATA
29
21
23
EXE
Import Address Table
22
24
A.DLL
B.DLL
Func AA Address
30
Func AB Address
31
Func AC Address
32
Func BA Address
33
Func BB Address
34
Func BC Address
35
25
A.DLL
Func AA Code
36
Func AB Code
37
Func AC Code
38
26
B.DLL
Func BA Code
39
Func BB Code
40
Func BC Code
41
27
DLL

FIG. 3

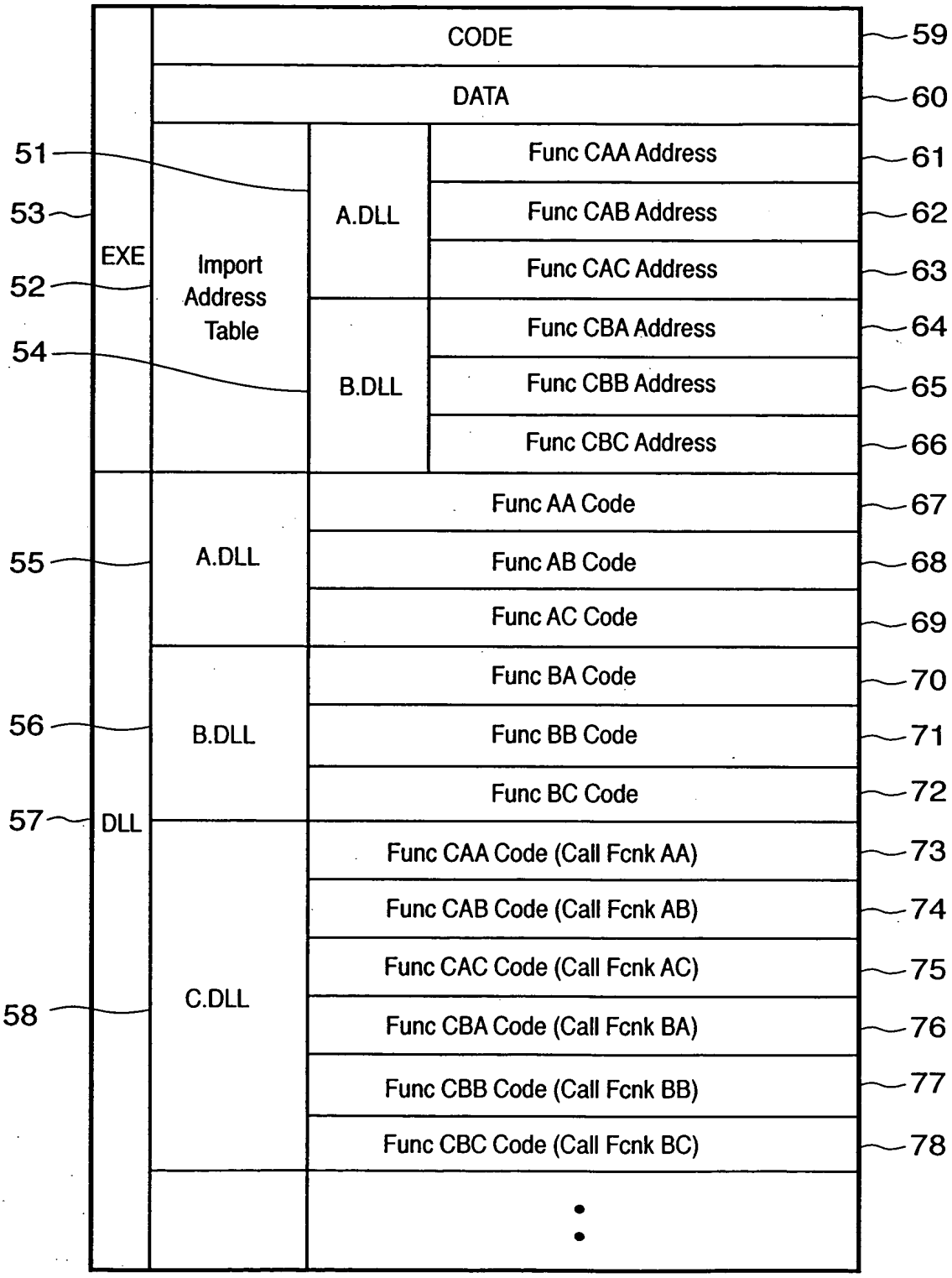

CODE
59
DATA
60
51
Func CAA Address
61
53
A.DLL
Func CAB Address
62
EXE
Import Address Table
Func CAC Address
63
52
Func CBA Address
64
54
B.DLL
Func CBB Address
65
Func CBC Address
66
Func AA Code
67
55
A.DLL
Func AB Code
68
Func AC Code
69
Func BA Code
70
56
B.DLL
Func BB Code
71
Func BC Code
72
57
DLL
Func CAA Code (Call Fcnk AA)
73
Func CAB Code (Call Fcnk AB)
74
Func CAC Code (Call Fcnk AC)
75
58
C.DLL
Func CBA Code (Call Fcnk BA)
76
Func CBB Code (Call Fcnk BB)
77
Func CBC Code (Call Fcnk BC)
78

FIG. 4A

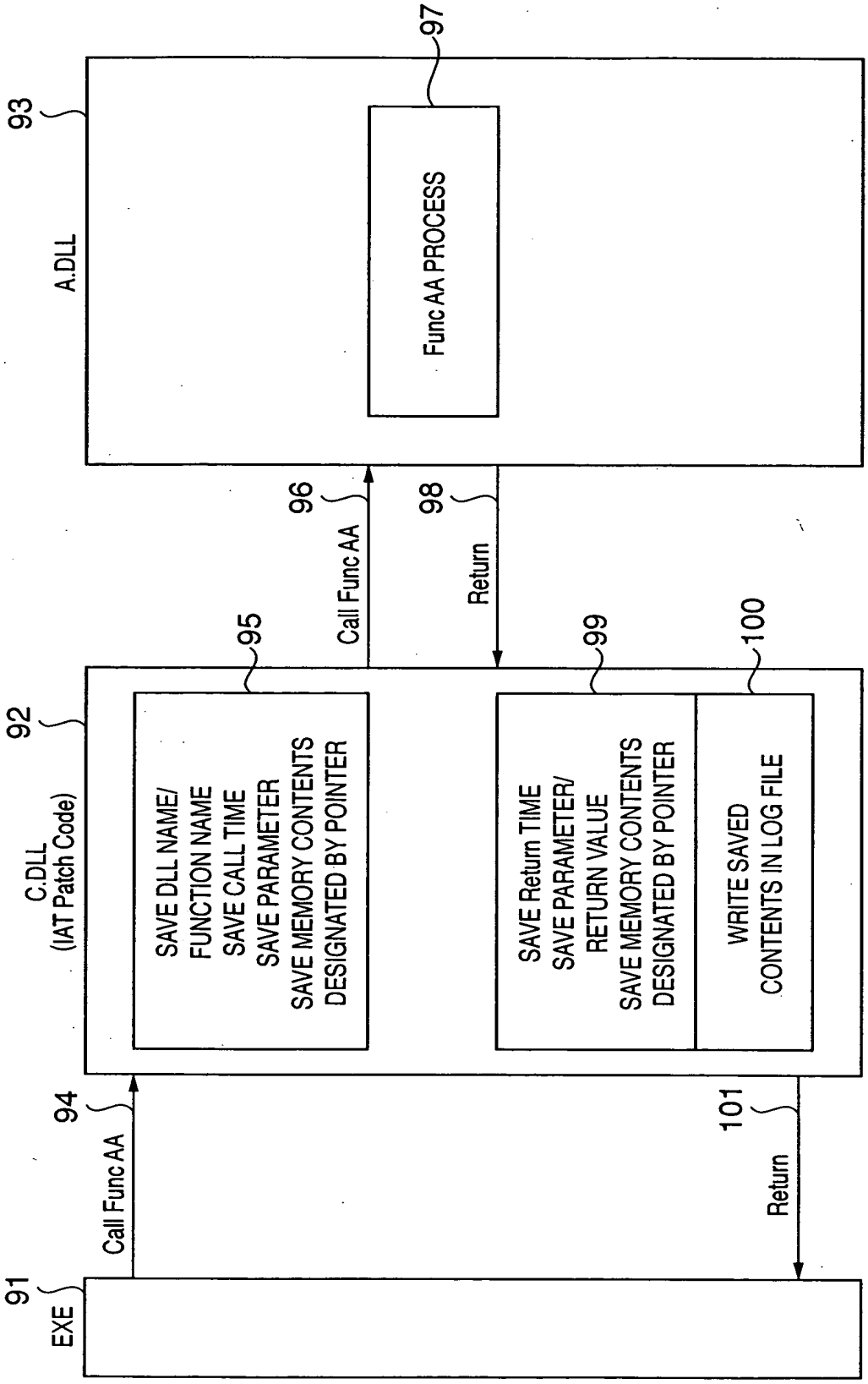

91
EXE
94
Call Func AA
92
C.DLL
(IAT Patch Code)
95
SAVE DLL NAME/
FUNCTION NAME
SAVE CALL TIME
SAVE PARAMETER
SAVE MEMORY CONTENTS
DESIGNATED BY POINTER
96
Call Func AA
93
A.DLL
97
Func AA PROCESS
98
Return
99
SAVE Return TIME
SAVE PARAMETER/
RETURN VALUE
SAVE MEMORY CONTENTS
DESIGNATED BY POINTER
100
WRITE SAVED
CONTENTS IN LOG FILE
101
Return

FIG. 4B

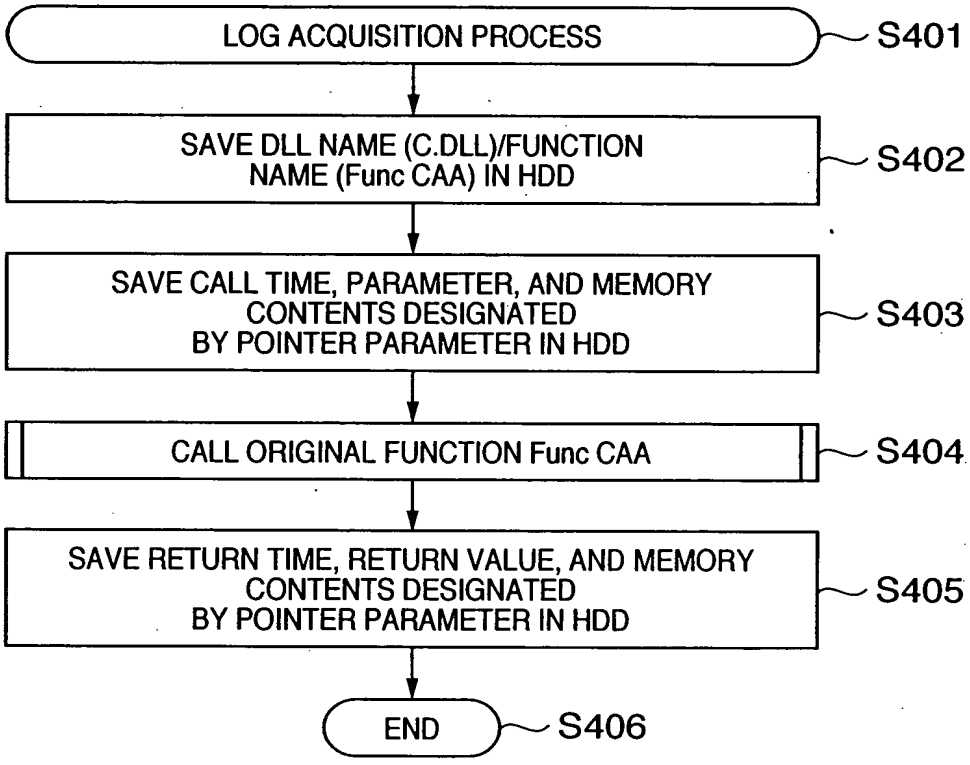

LOG ACQUISITION PROCESS
S401
SAVE DLL NAME (C.DLL)/FUNCTION NAME (Func CAA) IN HDD
S402
SAVE CALL TIME, PARAMETER, AND MEMORY CONTENTS DESIGNATED BY POINTER PARAMETER IN HDD
S403
CALL ORIGINAL FUNCTION Func CAA
S404
SAVE RETURN TIME, RETURN VALUE, AND MEMORY CONTENTS DESIGNATED BY POINTER PARAMETER IN HDD
S405
END
S406

FIG. 5

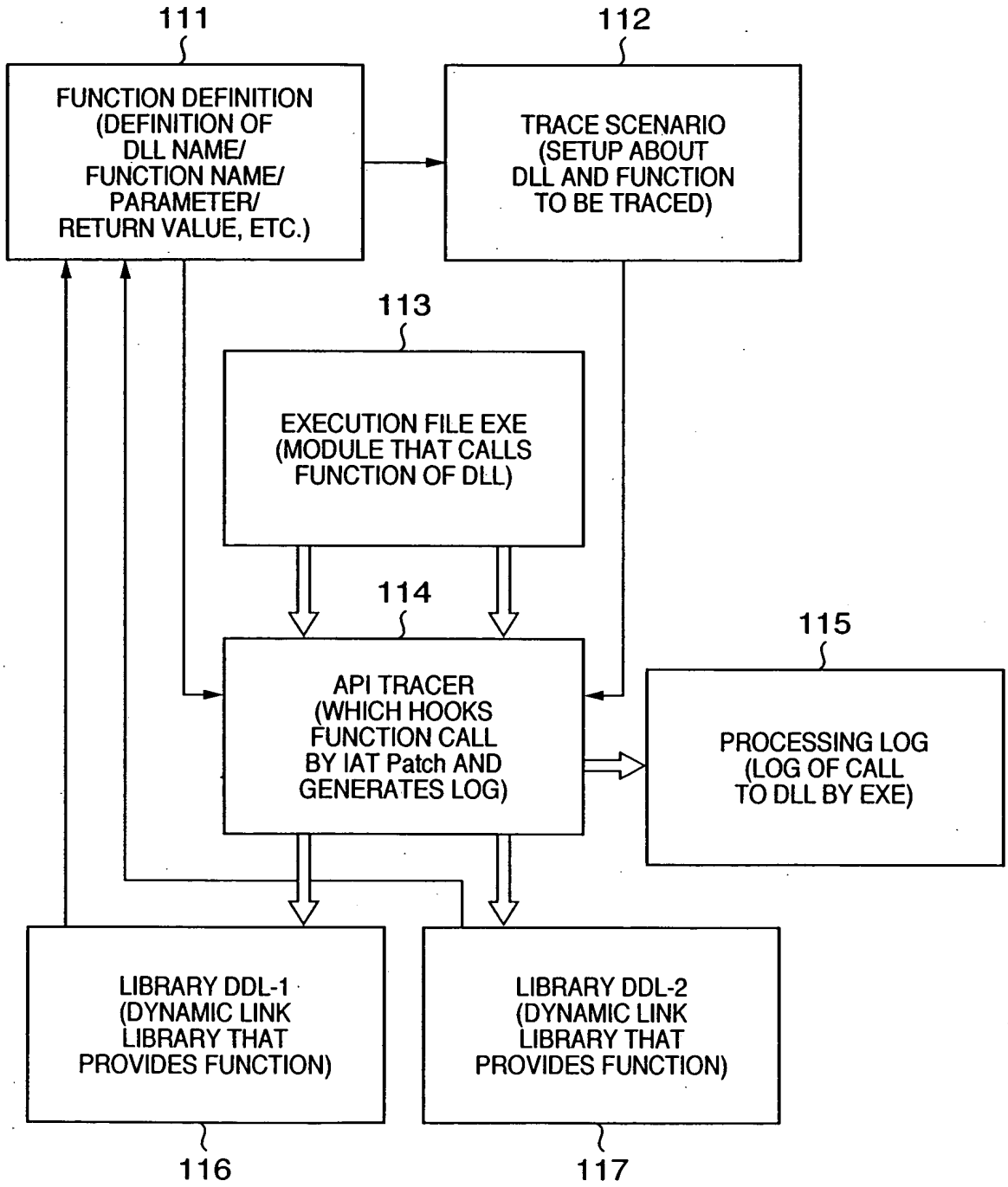

111
FUNCTION DEFINITION (DEFINITION OF DLL NAME/ FUNCTION NAME/ PARAMETER/ RETURN VALUE, ETC.)
112
TRACE SCENARIO (SETUP ABOUT DLL AND FUNCTION TO BE TRACED)
113
EXECUTION FILE EXE (MODULE THAT CALLS FUNCTION OF DLL)
114
API TRACER (WHICH HOOKS FUNCTION CALL BY IAT Patch AND GENERATES LOG)
115
PROCESSING LOG (LOG OF CALL TO DLL BY EXE)
LIBRARY DDL-1 (DYNAMIC LINK LIBRARY THAT PROVIDES FUNCTION)
116
LIBRARY DDL-2 (DYNAMIC LINK LIBRARY THAT PROVIDES FUNCTION)
117

FIG. 6

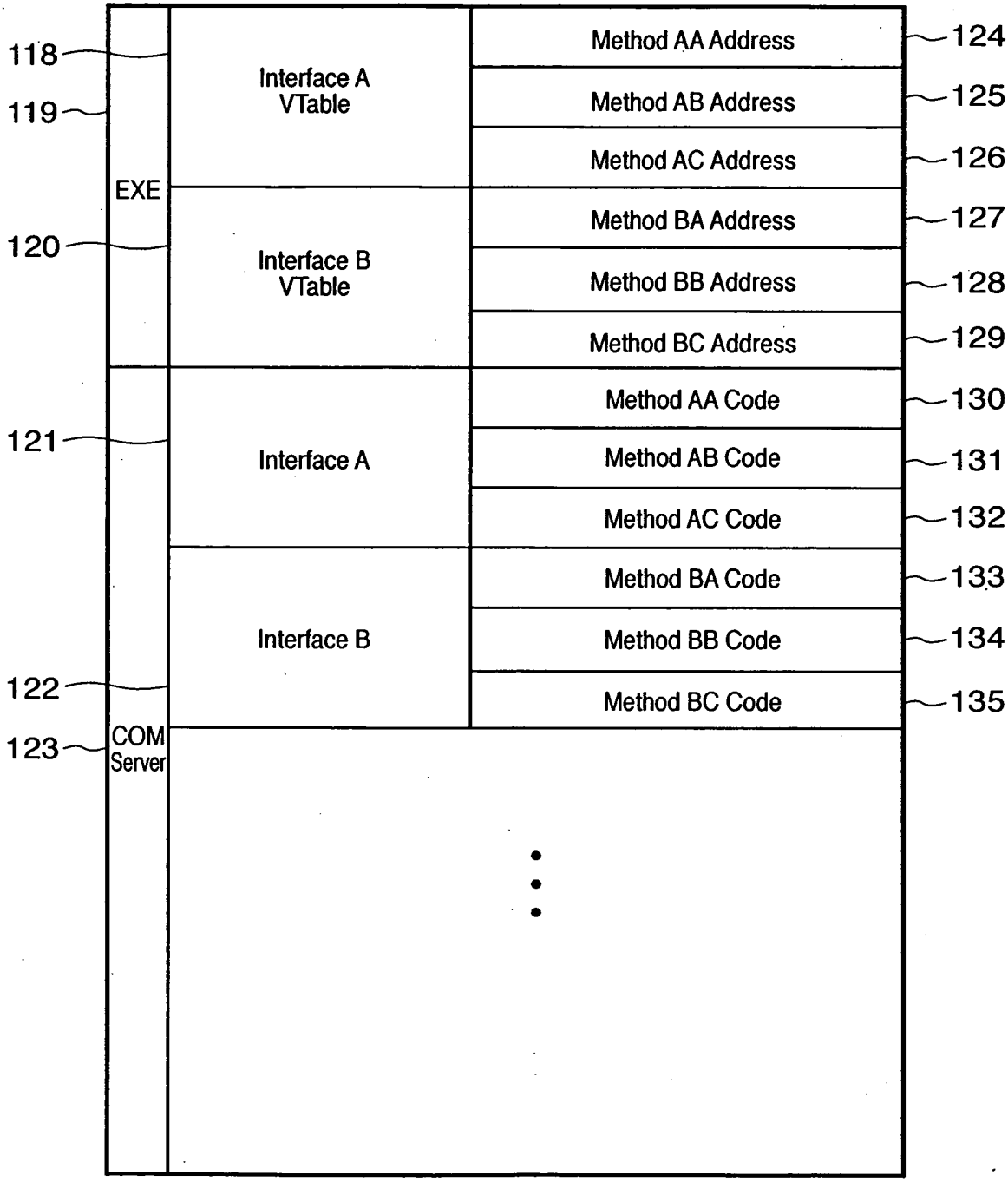

118
119
120
121
122
123
EXE
COM Server
Interface A VTable
Interface B VTable
Interface A
Interface B
Method AA Address
Method AB Address
Method AC Address
Method BA Address
Method BB Address
Method BC Address
Method AA Code
Method AB Code
Method AC Code
Method BA Code
Method BB Code
Method BC Code
124
125
126
127
128
129
130
131
132
133
134
135

FIG. 7

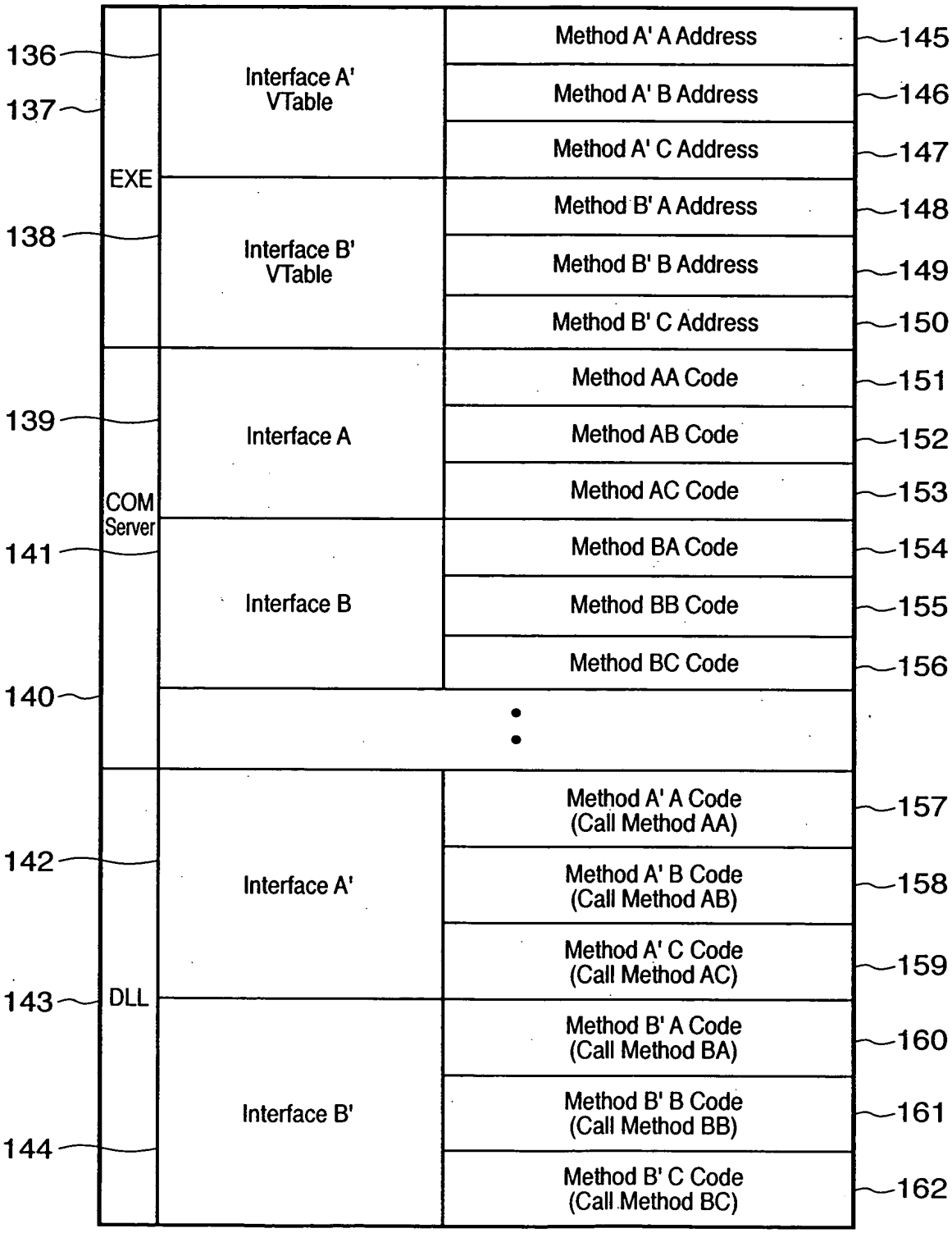

136
137
138
139
141
140
142
143
144
EXE
COM Server
DLL
Interface A' VTable
Interface B' VTable
Interface A
Interface B
Interface A'
Interface B'
Method A' A Address
Method A' B Address
Method A' C Address
Method B' A Address
Method B' B Address
Method B' C Address
Method AA Code
Method AB Code
Method AC Code
Method BA Code
Method BB Code
Method BC Code
Method A' A Code (Call Method AA)
Method A' B Code (Call Method AB)
Method A' C Code (Call Method AC)
Method B' A Code (Call Method BA)
Method B' B Code (Call Method BB)
Method B' C Code (Call Method BC)
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162

FIG. 8A

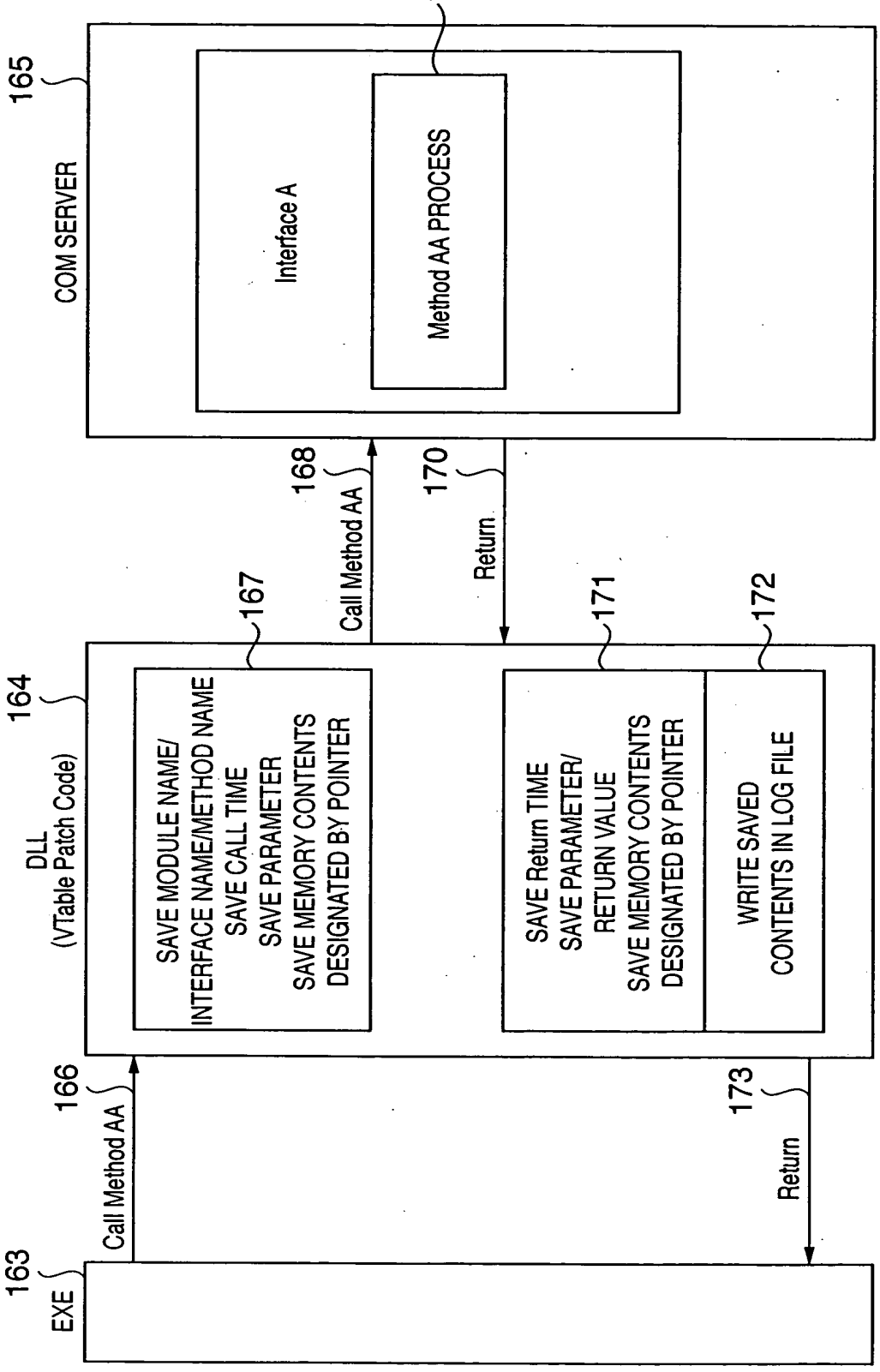

165
COM SERVER
169
Interface A
Method AA PROCESS
168
Call Method AA
170
Return
164
DLL
(VTable Patch Code)
167
SAVE MODULE NAME/
INTERFACE NAME/METHOD NAME
SAVE CALL TIME
SAVE PARAMETER
SAVE MEMORY CONTENTS
DESIGNATED BY POINTER
171
SAVE Return TIME
SAVE PARAMETER/
RETURN VALUE
SAVE MEMORY CONTENTS
DESIGNATED BY POINTER
172
WRITE SAVED
CONTENTS IN LOG FILE
166
Call Method AA
173
Return
163
EXE

FIG. 8B

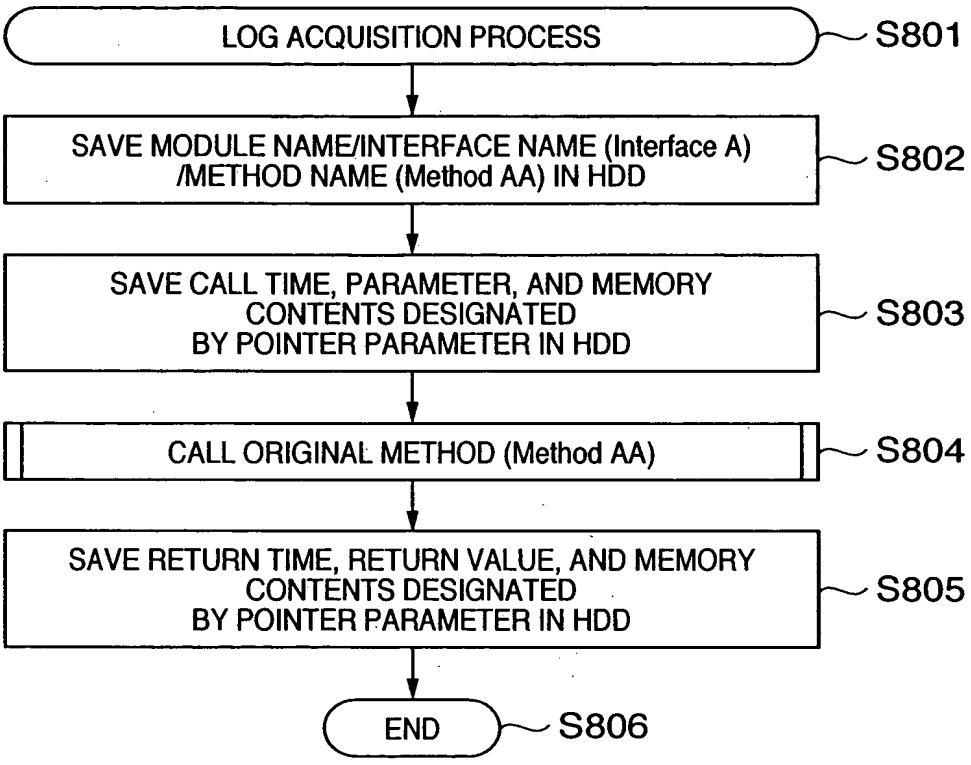

LOG ACQUISITION PROCESS
S801
SAVE MODULE NAME/INTERFACE NAME (Interface A) /METHOD NAME (Method AA) IN HDD
S802
SAVE CALL TIME, PARAMETER, AND MEMORY CONTENTS DESIGNATED BY POINTER PARAMETER IN HDD
S803
CALL ORIGINAL METHOD (Method AA)
S804
SAVE RETURN TIME, RETURN VALUE, AND MEMORY CONTENTS DESIGNATED BY POINTER PARAMETER IN HDD
S805
END
S806

FIG. 9

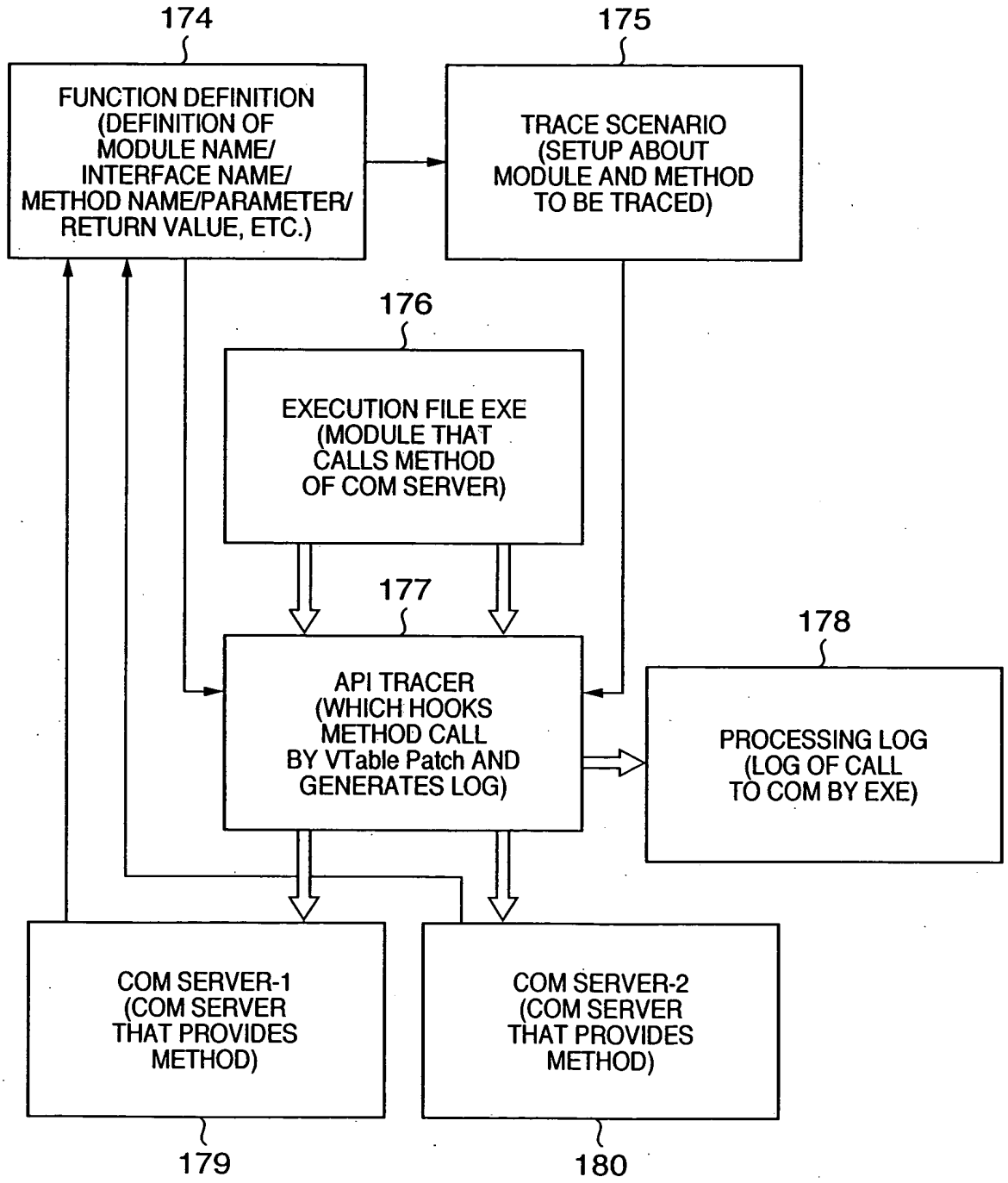

174
FUNCTION DEFINITION (DEFINITION OF MODULE NAME/ INTERFACE NAME/ METHOD NAME/PARAMETER/ RETURN VALUE, ETC.)
175
TRACE SCENARIO (SETUP ABOUT MODULE AND METHOD TO BE TRACED)
176
EXECUTION FILE EXE (MODULE THAT CALLS METHOD OF COM SERVER)
177
API TRACER (WHICH HOOKS METHOD CALL BY VTable Patch AND GENERATES LOG)
178
PROCESSING LOG (LOG OF CALL TO COM BY EXE)
COM SERVER-1 (COM SERVER THAT PROVIDES METHOD)
COM SERVER-2 (COM SERVER THAT PROVIDES METHOD)
179
180

FIG. 10

Interface NAME :	Interface A
Method NAME : Method AA	
ARGUMENT :	DWORD dwID
RETURN VALUE :	DWORD dwRet
Interface NAME :	Interface A
Method NAME : Method AB	
ARGUMENT :	DWORD dwSize
RETURN VALUE :	int nRet
Interface NAME :	Interface A
Method NAME : Method AC	
ARGUMENT :	DWORD dwSize
RETURN VALUE :	DWORD dwRet
Interface NAME :	Interface B
Method NAME : Method BA	
ARGUMENT :	DWORD dwSize
RETURN VALUE :	int nRet
Interface NAME :	Interface B
Method NAME : Method BC	
ARGUMENT :	DWORD dwID
RETURN VALUE :	DWORD dwRet

FIG. 11

Interface NAME : Interface A
Method NAME : Method AA
ARGUMENT : DWORD dwID : 256
RETURN VALUE : DWORD dwRet : 0
In TIME : 2002/03/25 22 : 24 : 12. 025
Out TIME : 2002/03/25 22 : 24 : 12. 035

Interface NAME : Interface B
Method NAME : Method BA
ARGUMENT : DWORD dwSize : 512
RETURN VALUE : int nRet : -1
In TIME : 2002/03/25 22 : 24 : 12. 046
Out TIME : 2002/03/25 22 : 24 : 12. 057

Interface NAME : Interface B
Method NAME : Method BC
ARGUMENT : DWORD dwID : 12
RETURN VALUE : DWORD dwRet : 2
In TIME : 2002/03/25 22 : 24 : 12. 068
Out TIME : 2002/03/25 22 : 24 : 12. 079

...

FIG. 12

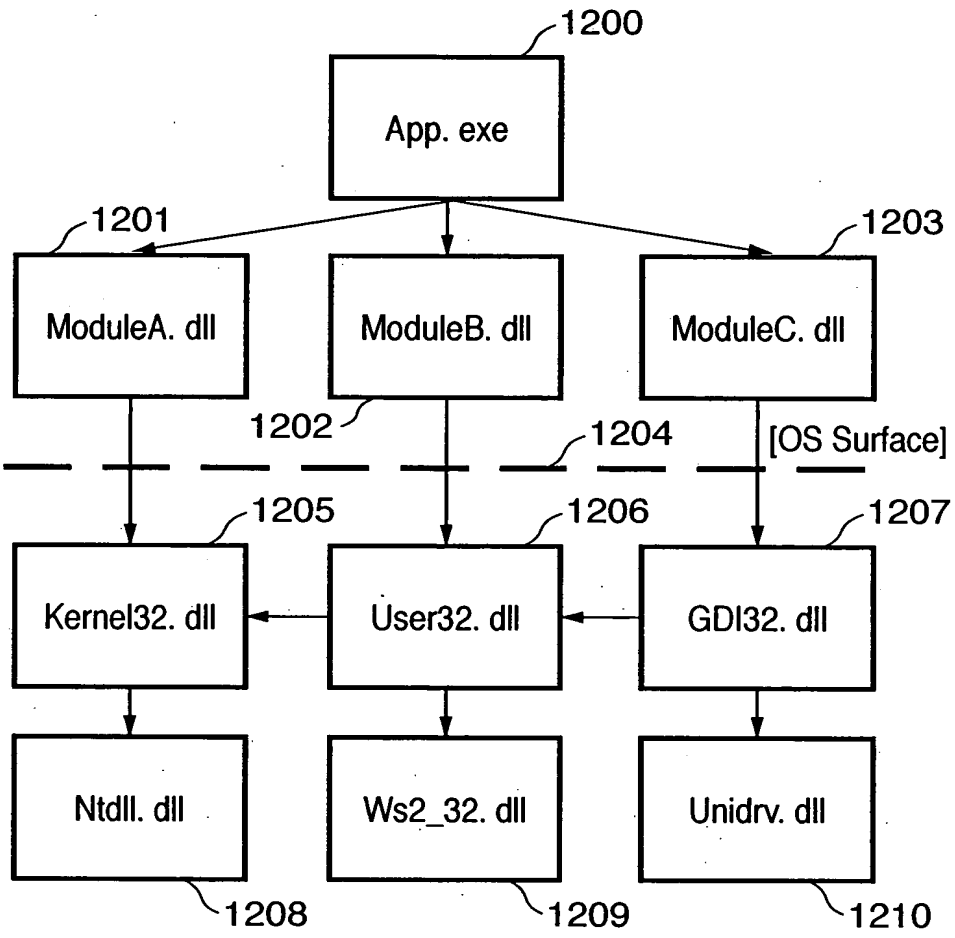

1200
App. exe
1201
1203
ModuleA. dll
ModuleB. dll
ModuleC. dll
1202
1204
[OS Surface]
1205
1206
1207
Kernel32. dll
User32. dll
GDI32. dll
Ntdll. dll
Ws2_32. dll
Unidrv. dll
1208
1209
1210

FIG. 13

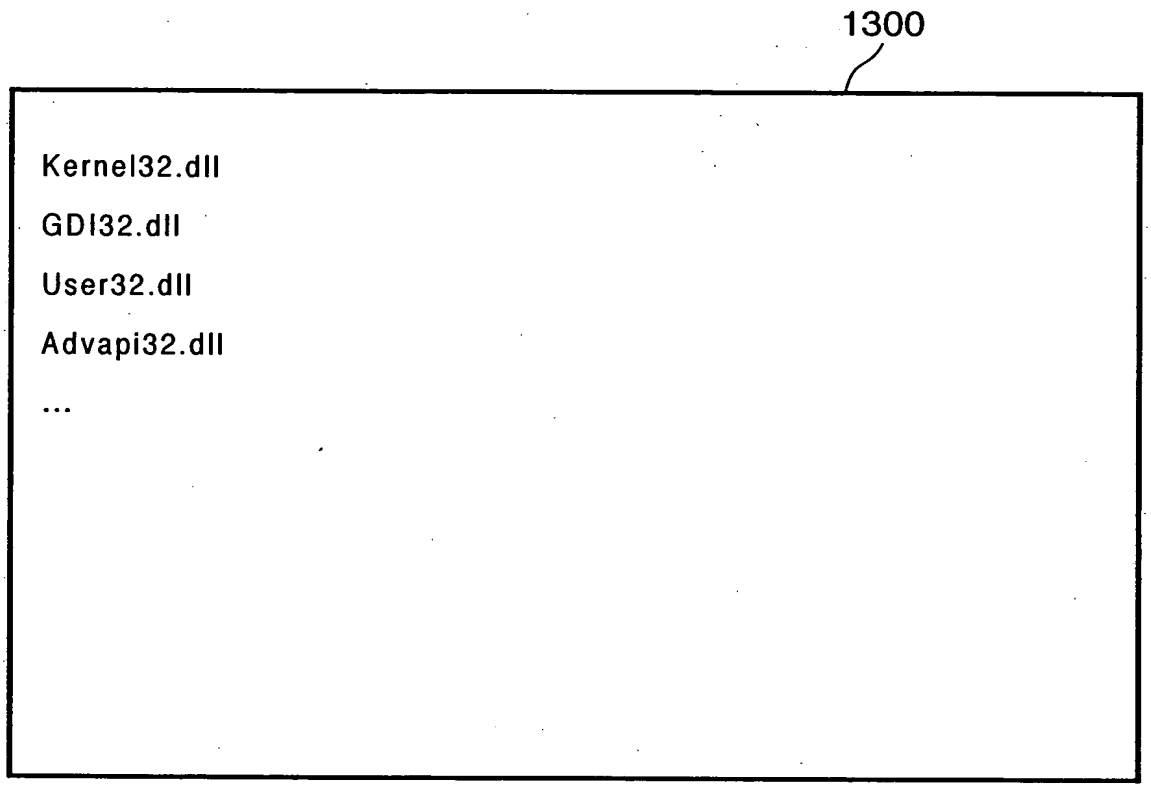


FIG. 14

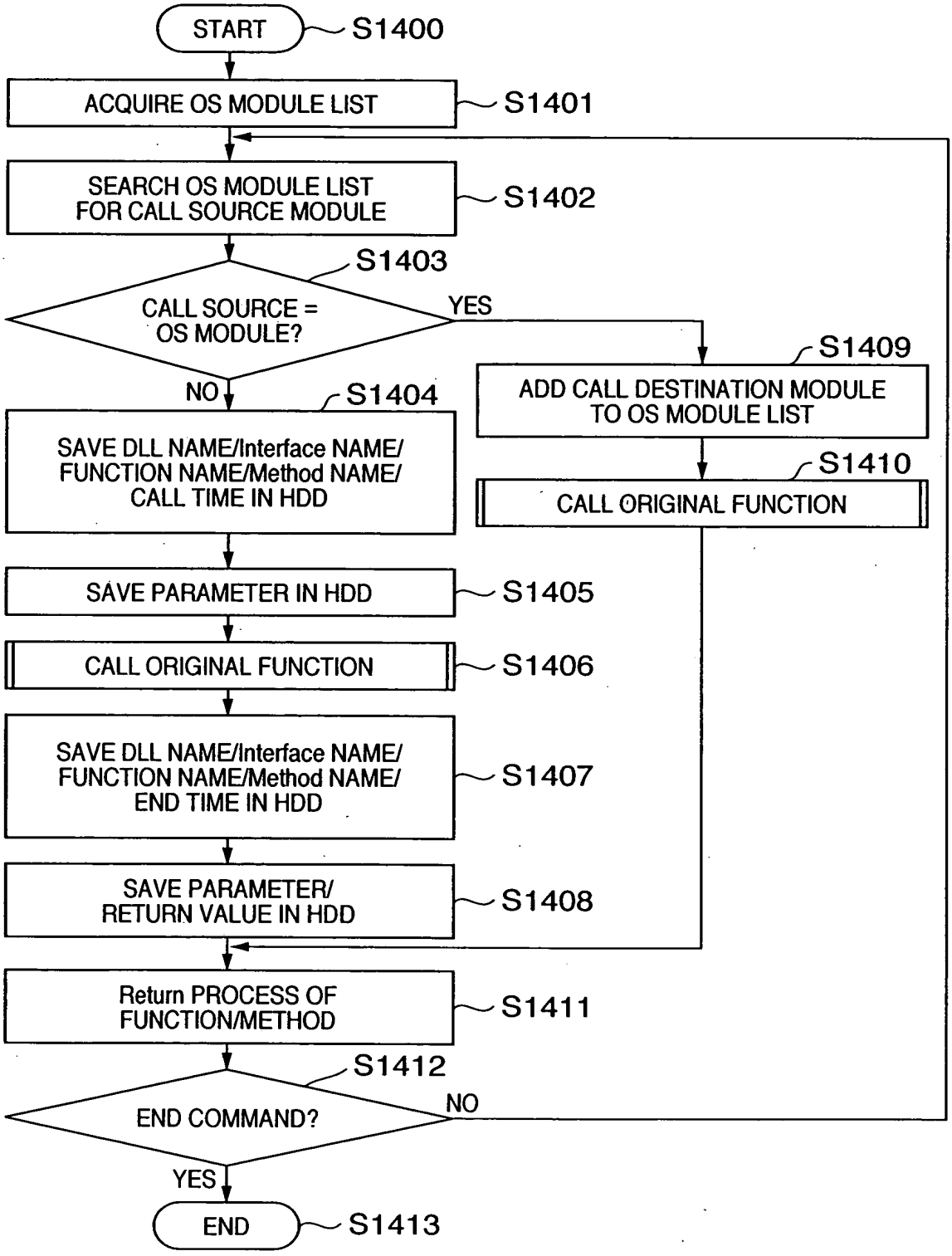

START
S1400
ACQUIRE OS MODULE LIST
S1401
SEARCH OS MODULE LIST FOR CALL SOURCE MODULE
S1402
S1403
CALL SOURCE = OS MODULE?
YES
NO
S1409
ADD CALL DESTINATION MODULE TO OS MODULE LIST
S1410
CALL ORIGINAL FUNCTION
S1404
SAVE DLL NAME/Interface NAME/ FUNCTION NAME/Method NAME/ CALL TIME IN HDD
SAVE PARAMETER IN HDD
S1405
CALL ORIGINAL FUNCTION
S1406
SAVE DLL NAME/Interface NAME/ FUNCTION NAME/Method NAME/ END TIME IN HDD
S1407
SAVE PARAMETER/ RETURN VALUE IN HDD
S1408
Return PROCESS OF FUNCTION/METHOD
S1411
S1412
END COMMAND?
NO
YES
END
S1413

FIG. 15A

1500

MODULE NAME	ModuleA. dll
FUNCTION NAME :	FuncA
In ARGUMENT :	int nIndex : 3
Out ARGUMENT :	DWORD* nRet : 0x007a61c0/0x0000000a(10)
RETURN VALUE :	int : 0
In TIME :	2002/03/25 22 : 24 : 12. 025
Out TIME :	2002/03/25 22 : 24 : 12. 035
MODULE NAME	Kernel32. dll
FUNCTION NAME :	FuncB
In ARGUMENT :	int nIndex : 1
Out ARGUMENT :	DWORD* nRet : 0x007a625c/0x0000000b(11)
RETURN VALUE :	int : 0
In TIME :	2002/03/25 22 : 24 : 12. 046
Out TIME :	2002/03/25 22 : 24 : 12. 057
MODULE NAME	ModuleB. dll
FUNCTION NAME :	FuncD
In ARGUMENT :	int nIndex : 4
Out ARGUMENT :	DWORD* nRet : 0x007b61c0/0x0000000d(13)
RETURN VALUE :	int : 0
In TIME :	2002/03/25 22 : 24 : 12. 089
Out TIME :	2002/03/25 22 : 24 : 13. 000
MODULE NAME	User32. dll
FUNCTION NAME :	FuncE
In ARGUMENT :	int nIndex : 6
Out ARGUMENT :	DWORD* nRet : 0x002a61c0/0x0000000e(14)
RETURN VALUE :	int : 0
In TIME :	2002/03/25 22 : 24 : 13. 011
Out TIME :	2002/03/25 22 : 24 : 13. 022

...

FIG. 15B

1300

```
MODULE NAME          ModuleA. dll
FUNCTION NAME :      FuncA
In ARGUMENT :        int nIndex :      3
Out ARGUMENT :       DWORD* nRet : 0x007a61c0/0x0000000a(10)
RETURN VALUE :       int :             0
In TIME :            2002/03/25 22 : 24 : 12. 025
Out TIME :           2002/03/25 22 : 24 : 12. 035

MODULE NAME          Kernel32. dll
FUNCTION NAME :      FuncB
In ARGUMENT :        int nIndex :      1
Out ARGUMENT :       DWORD* nRet : 0x007a625c/0x0000000b(11)
RETURN VALUE :       int :             0
In TIME :            2002/03/25 22 : 24 : 12. 046
Out TIME :           2002/03/25 22 : 24 : 12. 057

MODULE NAME          Mtdll. dll
FUNCTION NAME :      FuncC
In ARGUMENT :        int nIndex :      2
Out ARGUMENT :       DWORD* nRet : 0x007a6122/0x0000000c(12)
RETURN VALUE :       int :             0
In TIME :            2002/03/25 22 : 24 : 12. 068
Out TIME :           2002/03/25 22 : 24 : 12. 079

MODULE NAME          ModuleB. dll
FUNCTION NAME :      FuncD
In ARGUMENT :        int nIndex :      4
Out ARGUMENT :       DWORD* nRet : 0x007b61c0/0x0000000d(13)
RETURN VALUE :       int :             0
In TIME :            2002/03/25 22 : 24 : 12. 089
Out TIME :           2002/03/25 22 : 24 : 13. 000

MODULE NAME          User32. dll
FUNCTION NAME :      FuncE
In ARGUMENT :        int nIndex :      6
Out ARGUMENT :       DWORD* nRet : 0x002a61c0/0x0000000e(14)
RETURN VALUE :       int :             0
In TIME :            2002/03/25 22 : 24 : 13. 011
Out TIME :           2002/03/25 22 : 24 : 13. 022

MODULE NAME          Ws2_32. dll
FUNCTION NAME :      FuncF
In ARGUMENT :        int nIndex :      5
Out ARGUMENT :       DWORD* nRet : 0x002a6dc0/0x0000000f(15)
RETURN VALUE :       int :             0
In TIME :            2002/03/25 22 : 24 : 13. 034
Out TIME :           2002/03/25 22 : 24 : 13. 055
...
```

FIG. 16

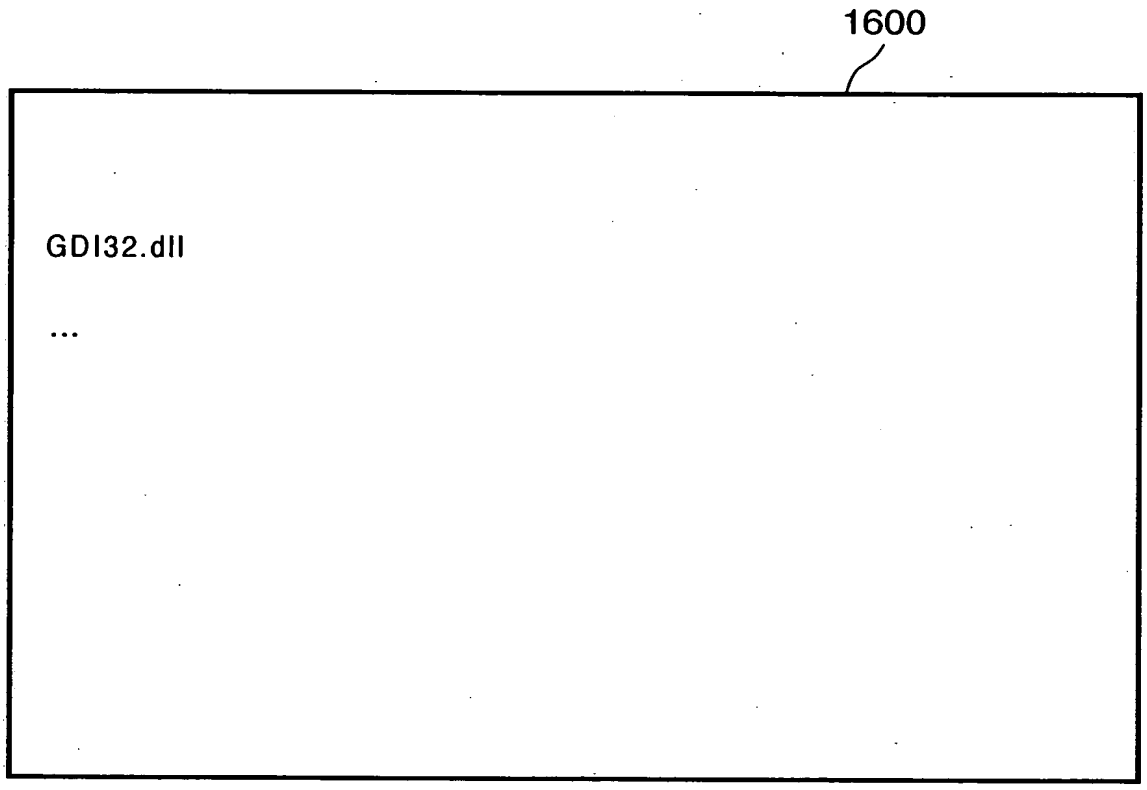


FIG. 17

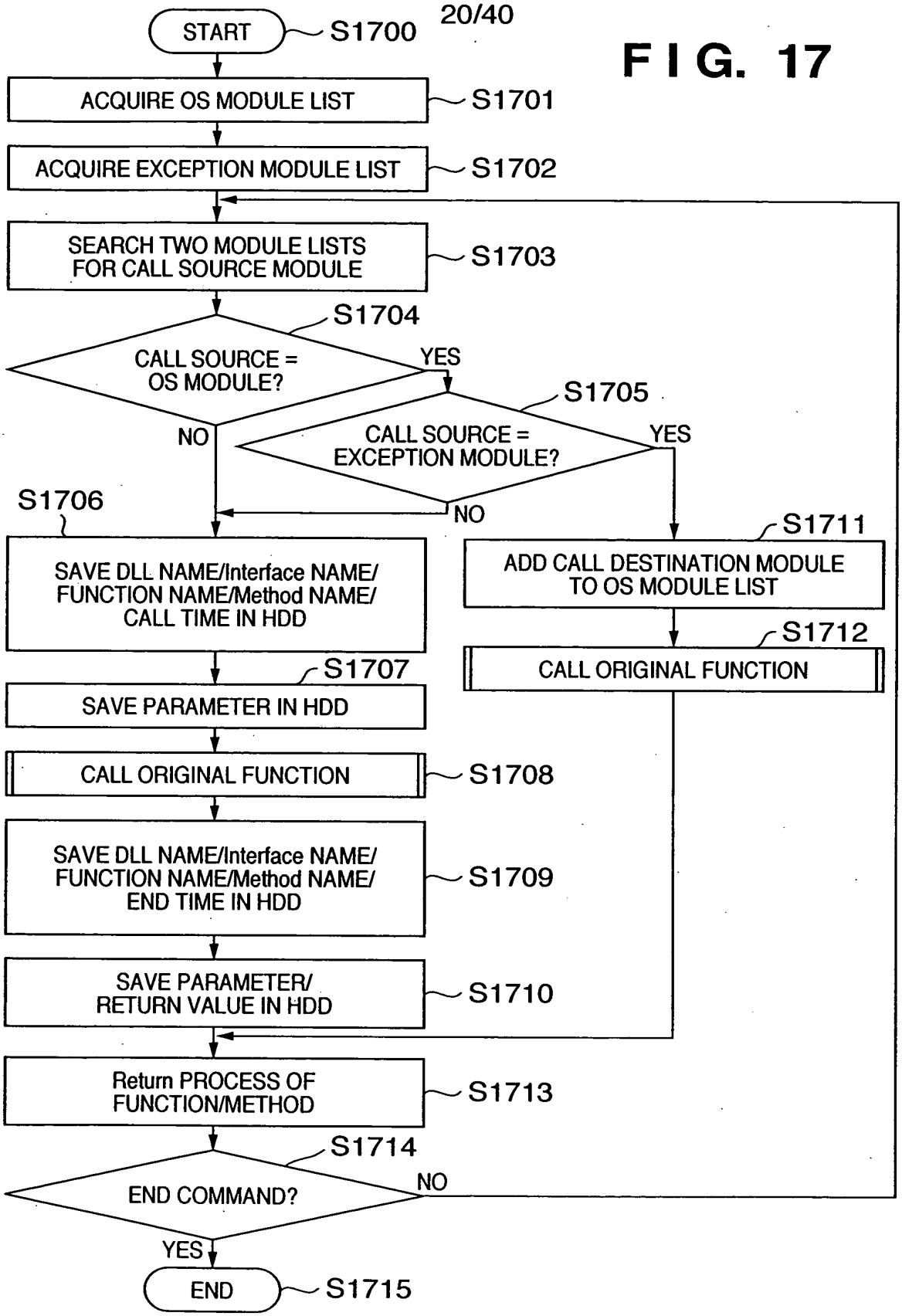

START
S1700
ACQUIRE OS MODULE LIST
S1701
ACQUIRE EXCEPTION MODULE LIST
S1702
SEARCH TWO MODULE LISTS FOR CALL SOURCE MODULE
S1703
S1704
CALL SOURCE = OS MODULE?
YES
NO
S1705
CALL SOURCE = EXCEPTION MODULE?
YES
NO
S1706
SAVE DLL NAME/Interface NAME/ FUNCTION NAME/Method NAME/ CALL TIME IN HDD
S1707
SAVE PARAMETER IN HDD
CALL ORIGINAL FUNCTION
S1708
SAVE DLL NAME/Interface NAME/ FUNCTION NAME/Method NAME/ END TIME IN HDD
S1709
SAVE PARAMETER/ RETURN VALUE IN HDD
S1710
S1711
ADD CALL DESTINATION MODULE TO OS MODULE LIST
S1712
CALL ORIGINAL FUNCTION
Return PROCESS OF FUNCTION/METHOD
S1713
S1714
END COMMAND?
NO
YES
END
S1715

FIG. 18


```
[
  uuid(58DB5633-0694-4340-97CE-4E1AC6BFFBA7),      //TestDllStd.
  helpstring("TestDllStd Type Library For PAT"),
  version(1.0)
]

library TestDllStd
{
        typedef [public] struct
        {
                char chParam;
                unsigned char uchParam;
                short sParam;
                unsigned short usParam;
                int nParam;
                unsigned int unParam;
                long lParam;
                unsigned long ulParam;
                double dbParam;
                float fParam;
        } TESTSTRUCT;
        typedef [public] TESTSTRUCT *LPTESTSTRUCT;

//DEFINE_GUID(GUID_PROGID, 0x8e037d65, 0xefa0, 0x40e7, 0x91, 0x43, 0xef,0x70, 0x56, 0x94, 0x5b,
0x79);
 [
  uuid(8E037D65-EFAO-40e7-9143-EF7056945B79),
  helpstring("TestDllStd.dll for PAT object." ),
 ]
        interface
        test
        {
                char _stdcall FuncCharStd([in] char chParam);
                char* _stdcall FuncPCharStd([in, out] char* lpchParam);

                TESTSTRUCT _stdcall FuncStructStd([in]TESTSTRUCT TestStruct);
                LPTESTSTRUCT _stdcall FuncPStructStd([in, out]LPTESTSTRUCT lpTestStruct);
        };
}
```

FIG. 19

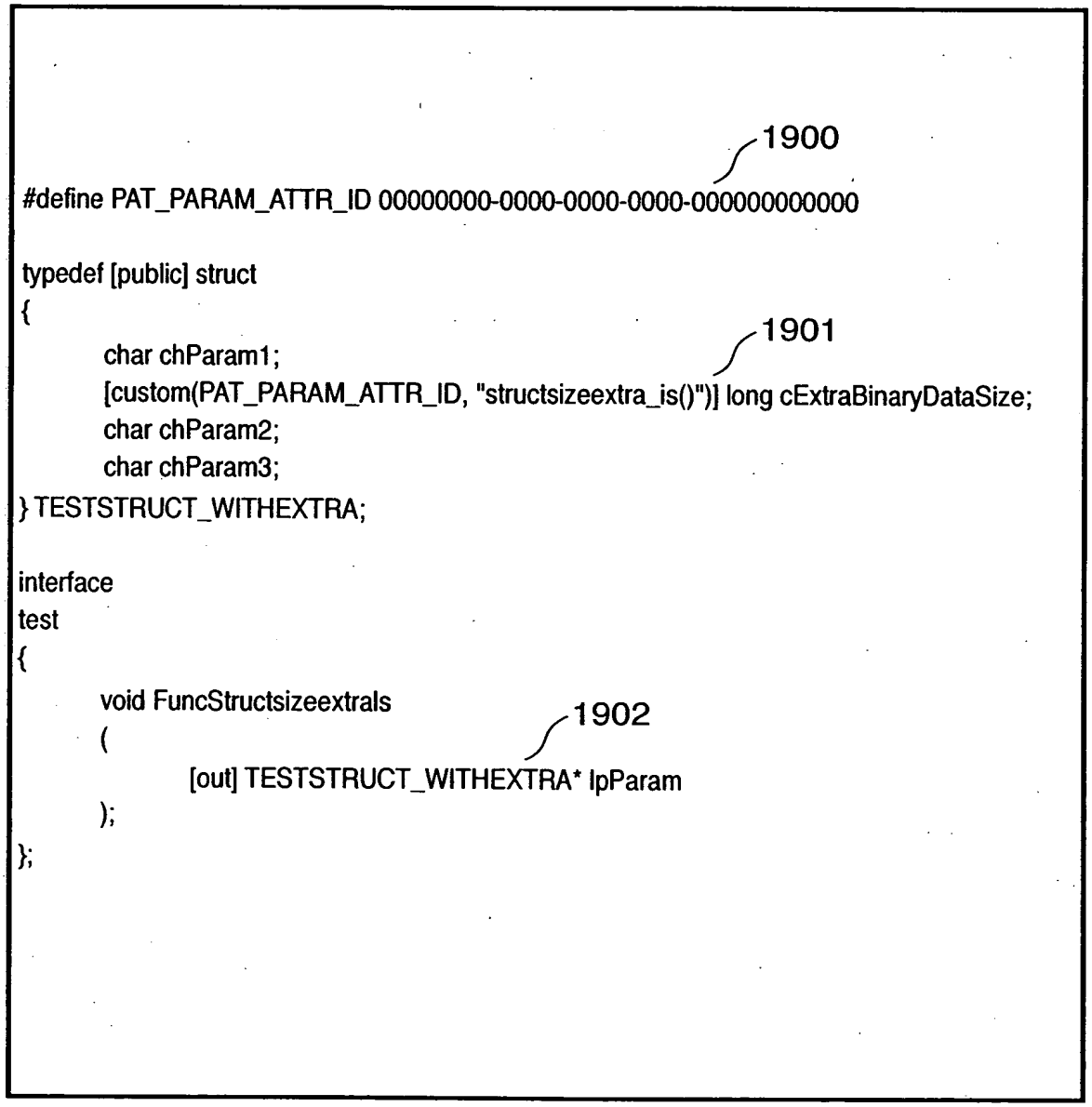

1900
#define PAT_PARAM_ATTR_ID 00000000-0000-0000-0000-000000000000
typedef [public] struct
{
char chParam1;
1901
[custom(PAT_PARAM_ATTR_ID, "structsizeextra_is()")] long cExtraBinaryDataSize;
char chParam2;
char chParam3;
} TESTSTRUCT_WITHEXTRA;
interface
test
{
void FuncStructsizeextrals
1902
(
[out] TESTSTRUCT_WITHEXTRA* lpParam
);
};

FIG. 20

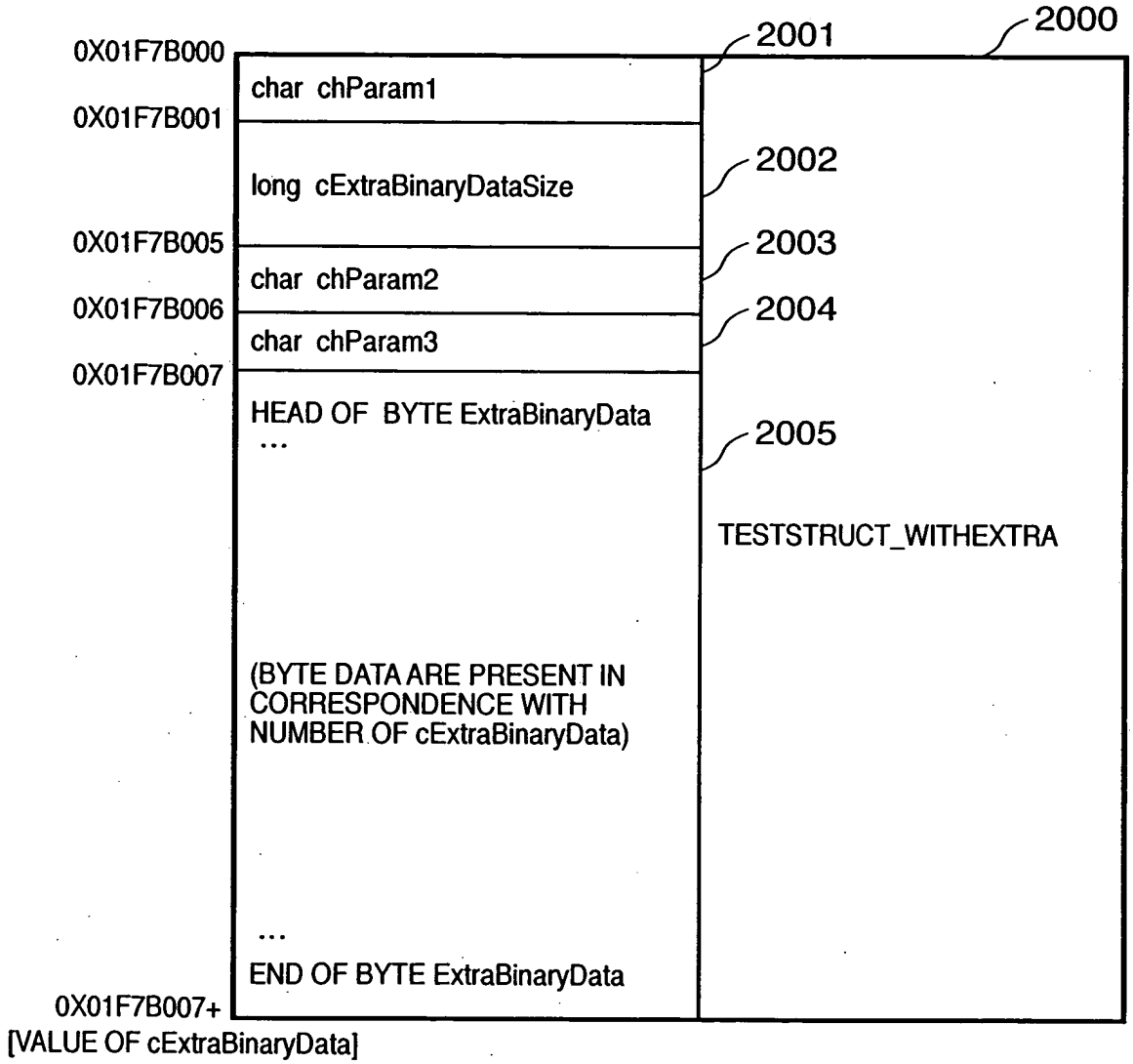

2000
2001
0X01F7B000
char chParam1
0X01F7B001
long cExtraBinaryDataSize
2002
0X01F7B005
2003
char chParam2
0X01F7B006
2004
char chParam3
0X01F7B007
HEAD OF BYTE ExtraBinaryData
...
2005
TESTSTRUCT_WITHEXTRA
(BYTE DATA ARE PRESENT IN CORRESPONDENCE WITH NUMBER OF cExtraBinaryData)
...
END OF BYTE ExtraBinaryData
0X01F7B007+
[VALUE OF cExtraBinaryData]

FIG. 21

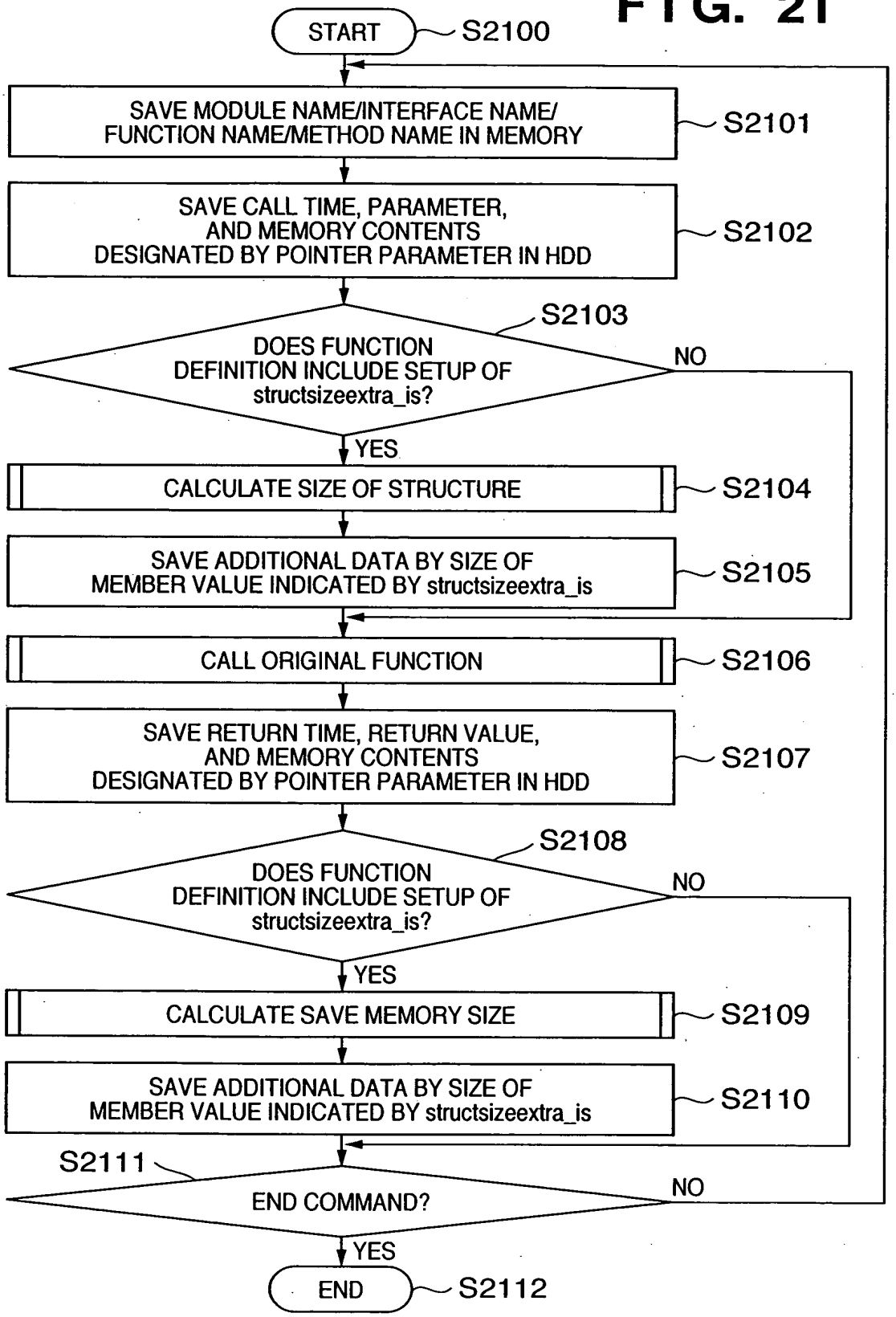

START
S2100
SAVE MODULE NAME/INTERFACE NAME/ FUNCTION NAME/METHOD NAME IN MEMORY
S2101
SAVE CALL TIME, PARAMETER, AND MEMORY CONTENTS DESIGNATED BY POINTER PARAMETER IN HDD
S2102
S2103
DOES FUNCTION DEFINITION INCLUDE SETUP OF structsizeextra_is?
NO
YES
CALCULATE SIZE OF STRUCTURE
S2104
SAVE ADDITIONAL DATA BY SIZE OF MEMBER VALUE INDICATED BY structsizeextra_is
S2105
CALL ORIGINAL FUNCTION
S2106
SAVE RETURN TIME, RETURN VALUE, AND MEMORY CONTENTS DESIGNATED BY POINTER PARAMETER IN HDD
S2107
S2108
DOES FUNCTION DEFINITION INCLUDE SETUP OF structsizeextra_is?
NO
YES
CALCULATE SAVE MEMORY SIZE
S2109
SAVE ADDITIONAL DATA BY SIZE OF MEMBER VALUE INDICATED BY structsizeextra_is
S2110
S2111
END COMMAND?
NO
YES
END
S2112

FIG. 22

2200

```
MODULE NAME :     TestDllStd.DLL
FUNCTION NAME :   Func Structsizeextrals
ARGUMENT (in) :
ARGUMENT (out) :  struct TESTSTRUCT_WITHEXTRA :
                    char chParam1 : "a"
                    long cExtraBinaryDataSize: 8
                    char chParam2 : "b"
                    char chParam3 : "c"
                    ExtraBinaryData : 0x01F7B007. Data ID=0x0001
RETURN VALUE :    void :
In TIME :         2002/03/25 22 : 24 : 12. 025
Out TIME :        2002/03/25 22 : 24 : 12. 035
MODULE NAME :     TestDllStd.DLL
FUNCTION NAME :   Func Structsizeextrals
ARGUMENT (in) :
ARGUMENT (out) :  struct TESTSTRUCT_WITHEXTRA :
                    char chParam1 : "d"
                    long cExtraBinaryDataSize: 40
                    char chParam2 : "e"
                    char chParam3 : "f"
                    ExtraBinaryData : 0x01F7B007. Data ID=0x0002
RETURN VALUE :    void :
In TIME :         2002/03/25 22 : 24 : 12. 046
Out TIME :        2002/03/25 22 : 24 : 12. 057
MODULE NAME :     TestDllStd.DLL
FUNCTION NAME :   Func Structsizeextrals
ARGUMENT (in) :
ARGUMENT (out) :  struct TESTSTRUCT_WITHEXTRA :
                    char chParam1 : "g"
                    long cExtraBinaryDataSize: 5
                    char chParam2 : "h"
                    char chParam3 : "i"
                    ExtraBinaryData : 0x01F7B007. Data ID=0x0003
RETURN VALUE :    void :
In TIME :         2002/03/25 22 : 24 : 12. 068
Out TIME :        2002/03/25 22 : 24 : 12. 079
MODULE NAME :     TestDllStd.DLL
FUNCTION NAME :   Func Structsizeextrals
ARGUMENT (in) :
ARGUMENT (out) :  struct TESTSTRUCT_WITHEXTRA :
                    char chParam1 : "j"
                    long cExtraBinaryDataSize: 7
                    char chParam2 : "k"
                    char chParam3 : "l"
                    ExtraBinaryData : 0x01F7B007. Data ID=0x0004
RETURN VALUE :    void :
In TIME :         2002/03/25 22 : 24 : 12. 100
Out TIME :        2002/03/25 22 : 24 : 12. 179
...
```

2201

```
Data ID: 0x0001
Size: 8

00000000: 10 00 00 00 4A 03 A5 20

Data ID: 0x0002
Size: 40

00000000: 05 00 00 00 4A 03 A5 20
00000008: 05 00 00 00 4B 03 A5 20
00000010: 05 00 00 00 4C 03 A5 20
00000018: 05 00 00 00 4D 03 A5 20
00000020: 05 00 00 00 4E 03 A5 20

Data ID: 0x0003
Size: 40

00000000: 66 4A 70 50 00

Data ID: 0x0004
Size: 7

00000000: 01 5D 66 B2 20 49 20

...
```

FIG. 23

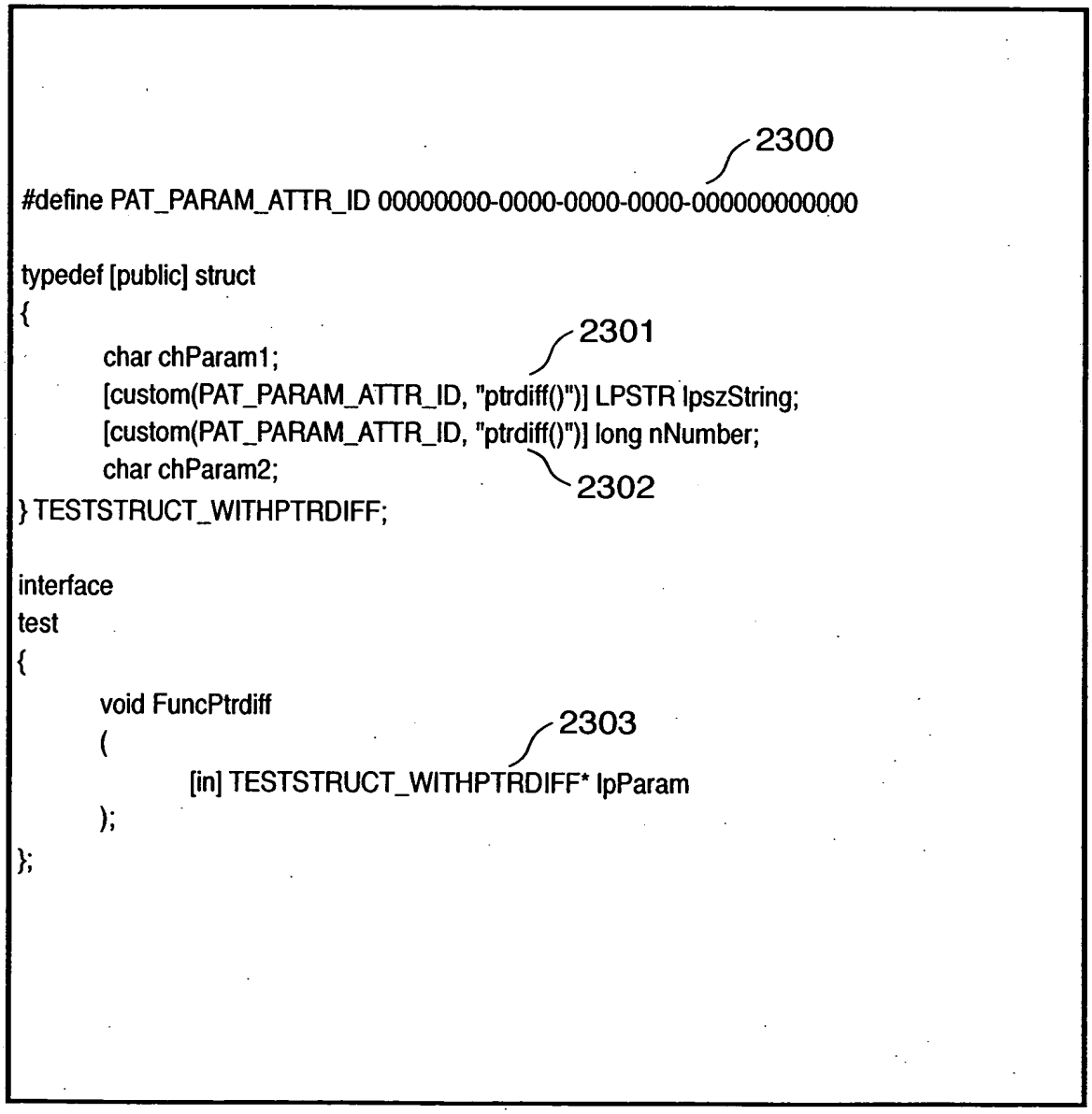

2300
#define PAT_PARAM_ATTR_ID 00000000-0000-0000-0000-000000000000
typedef [public] struct
{
char chParam1;
2301
[custom(PAT_PARAM_ATTR_ID, "ptrdiff()")] LPSTR lpszString;
[custom(PAT_PARAM_ATTR_ID, "ptrdiff()")] long nNumber;
char chParam2;
2302
} TESTSTRUCT_WITHPTRDIFF;
interface
test
{
void FuncPtrdiff
(
2303
[in] TESTSTRUCT_WITHPTRDIFF* lpParam
);
};

FIG. 24

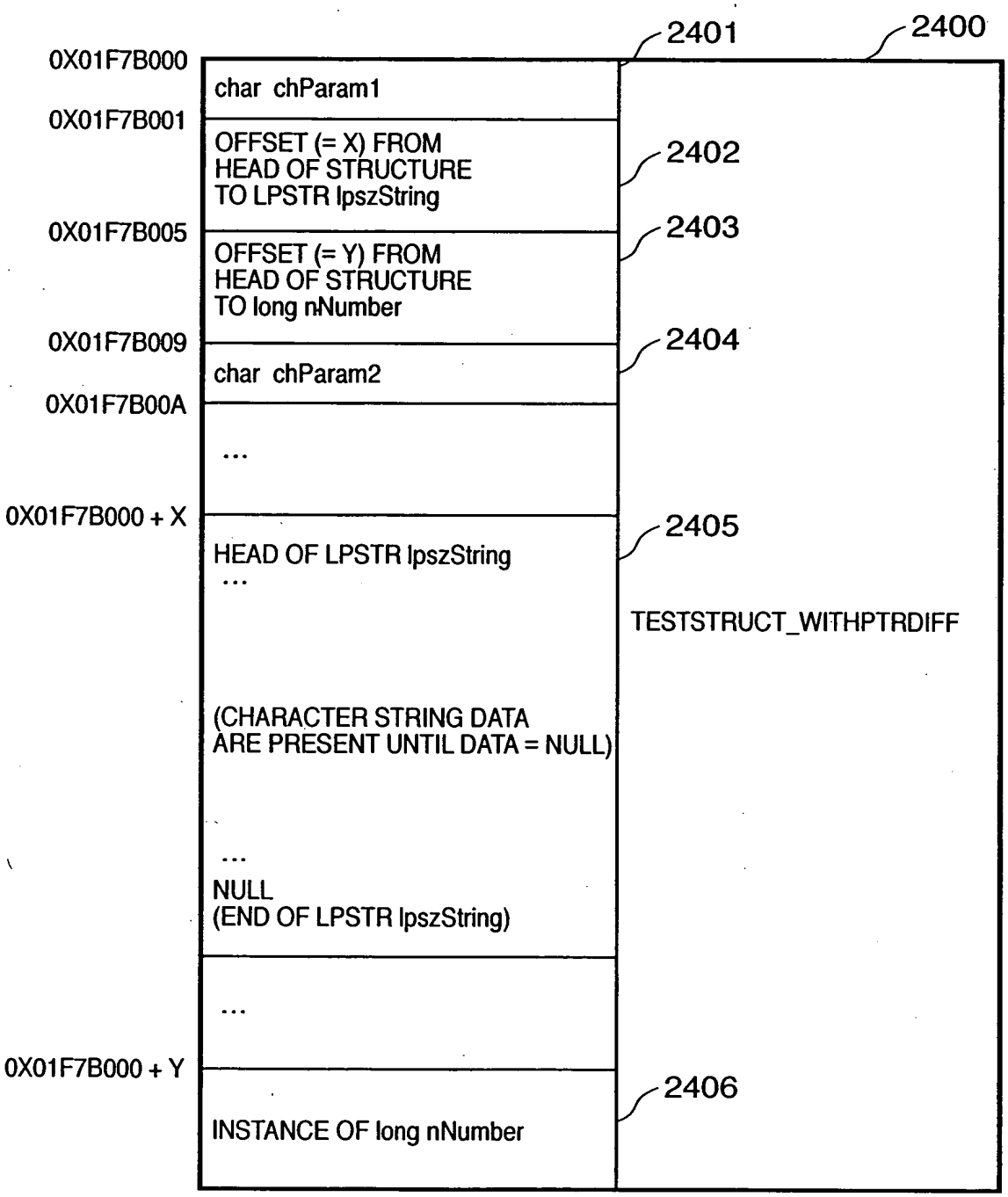

2401
2400
0X01F7B000
char chParam1
0X01F7B001
OFFSET (= X) FROM
HEAD OF STRUCTURE
TO LPSTR lpszString
2402
0X01F7B005
OFFSET (= Y) FROM
HEAD OF STRUCTURE
TO long nNumber
2403
0X01F7B009
char chParam2
2404
0X01F7B00A
...
0X01F7B000 + X
HEAD OF LPSTR lpszString
...
2405
TESTSTRUCT_WITHPTRDIFF
(CHARACTER STRING DATA
ARE PRESENT UNTIL DATA = NULL)
...
NULL
(END OF LPSTR lpszString)
...
0X01F7B000 + Y
INSTANCE OF long nNumber
2406

FIG. 25

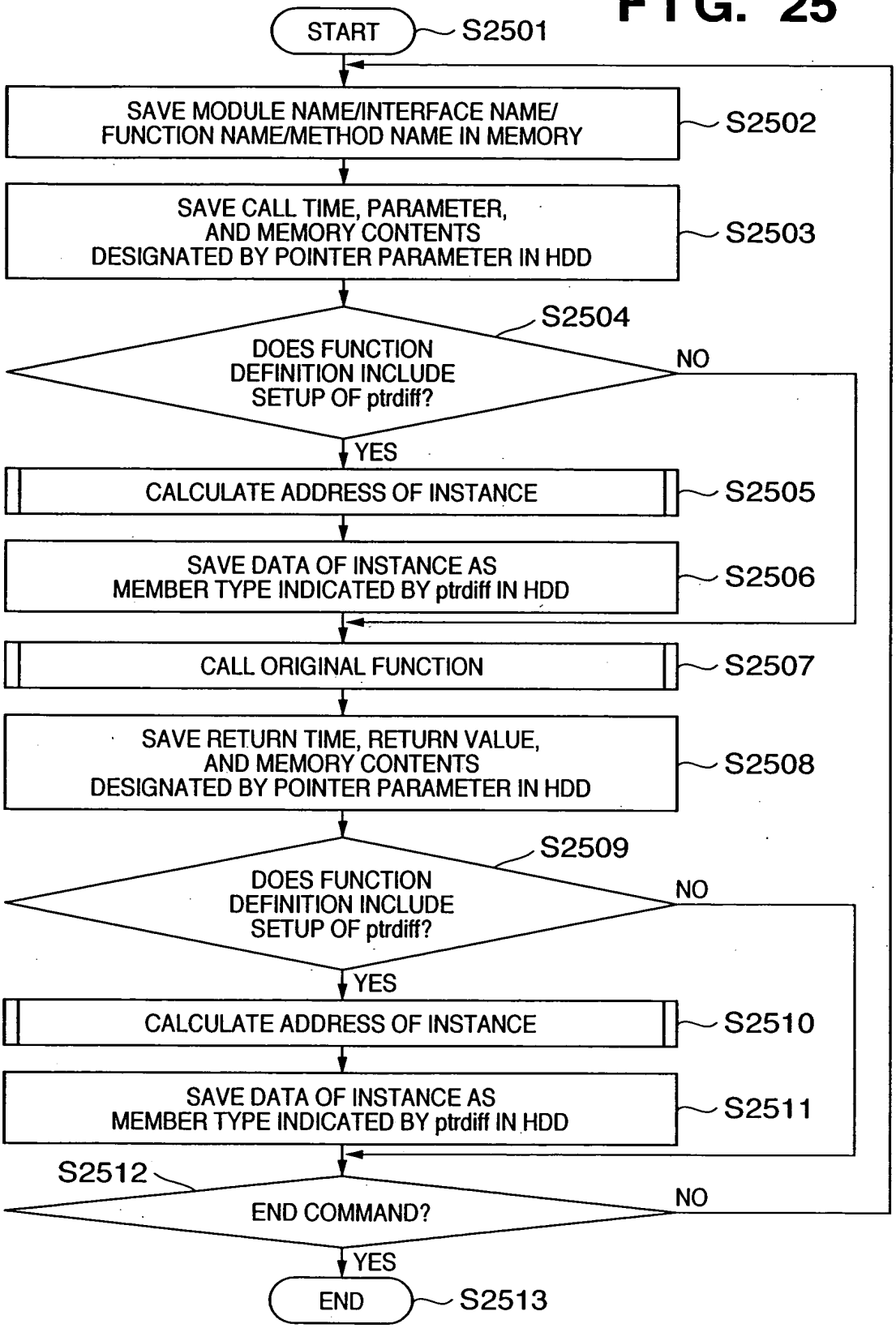

START
S2501
SAVE MODULE NAME/INTERFACE NAME/ FUNCTION NAME/METHOD NAME IN MEMORY
S2502
SAVE CALL TIME, PARAMETER, AND MEMORY CONTENTS DESIGNATED BY POINTER PARAMETER IN HDD
S2503
S2504
DOES FUNCTION DEFINITION INCLUDE SETUP OF ptrdiff?
NO
YES
CALCULATE ADDRESS OF INSTANCE
S2505
SAVE DATA OF INSTANCE AS MEMBER TYPE INDICATED BY ptrdiff IN HDD
S2506
CALL ORIGINAL FUNCTION
S2507
SAVE RETURN TIME, RETURN VALUE, AND MEMORY CONTENTS DESIGNATED BY POINTER PARAMETER IN HDD
S2508
S2509
DOES FUNCTION DEFINITION INCLUDE SETUP OF ptrdiff?
NO
YES
CALCULATE ADDRESS OF INSTANCE
S2510
SAVE DATA OF INSTANCE AS MEMBER TYPE INDICATED BY ptrdiff IN HDD
S2511
S2512
END COMMAND?
NO
YES
END
S2513

FIG. 26

2600

```
MODULE NAME :                TestDllStd.DLL
FUNCTION NAME :              FuncPtrdiff
ARGUMENT (in) :              struct TESTSTRUCT_WITHPTRDIFF :
                               char chParam1 : " a"
                               Offset to LPSTR lpszString : 16
                               LPSTR lpszString : "Test1"
                               Offset to long nNumber : 28
                               long nNumber : 1
                               char chParam3 : " b"
ARGUMENT (out) :
RETURN VALUE :               void :
In TIME :                    2002/03/25 22 : 24 : 12. 025
Out TIME :                   2002/03/25 22 : 24 : 12. 035
MODULE NAME :                TestDllStd.DLL
FUNCTION NAME :              FuncPtrdiff
ARGUMENT (in) :              struct TESTSTRUCT_WITHPTRDIFF :
                               char chParam1 : " c"
                               Offset to LPSTR lpszString : 18
                               LPSTR lpszString : "Test2"
                               Offset to long nNumber : 30
                               long nNumber : 2
                               char chParam3 : " d"
ARGUMENT (out) :
RETURN VALUE :               void :
In TIME :                    2002/03/25 22 : 24 : 12. 046
Out TIME :                   2002/03/25 22 : 24 : 12. 057
MODULE NAME :                TestDllStd.DLL
FUNCTION NAME :              FuncPtrdiff
ARGUMENT (in) :              struct TESTSTRUCT_WITHPTRDIFF :
                               char chParam1 : " e"
                               Offset to LPSTR lpszString : 20
                               LPSTR lpszString : "Test3"
                               Offset to long nNumber : 32
                               long nNumber : 3
                               char chParam3 : " f"
ARGUMENT (out) :
RETURN VALUE :               void :
In TIME :                    2002/03/25 22 : 24 : 12. 068
Out TIME :                   2002/03/25 22 : 24 : 12. 079
MODULE NAME :                TestDllStd.DLL
FUNCTION NAME :              FuncPtrdiff
ARGUMENT (in) :              struct TESTSTRUCT_WITHPTRDIFF :
                               char chParam1 : "g"
                               Offset to LPSTR lpszString : 16
                               LPSTR lpszString : "Test4"
                               Offset to long nNumber : 28
                               long nNumber : 29
                               char chParam3 : " h"
ARGUMENT (out) :
RETURN VALUE :               void :
In TIME :                    2002/03/25 22 : 24 : 12. 100
Out TIME :                   2002/03/25 22 : 24 : 12. 179
...
```

FIG. 27

```
                                                              2700
#define PAT_PARAM_ATTR_ID 00000000-0000-0000-0000-000000000000

interface test ;
                                  2701
typedef struct tagGUID {
        unsigned long Data1 ;
        unsigned short Data2 ;
        unsigned short Data3 ;
        unsigned char Data4[ 8 ] ;
} GUID ;

interface test {
        HRESULT _stdcall DllGetClassObject (                  2702
                [ in, custom(PAT_PARAM_ATTR_ID, "clsid_( )" ) ] GUID* rctsid,
                [ in ] GUID* riid,
                [ out, custom(PAT_PARAM_ATTR_ID, "iid_is(riid)" ) ] void** ppv) ;
};
                                                              2703
```

FIG. 28

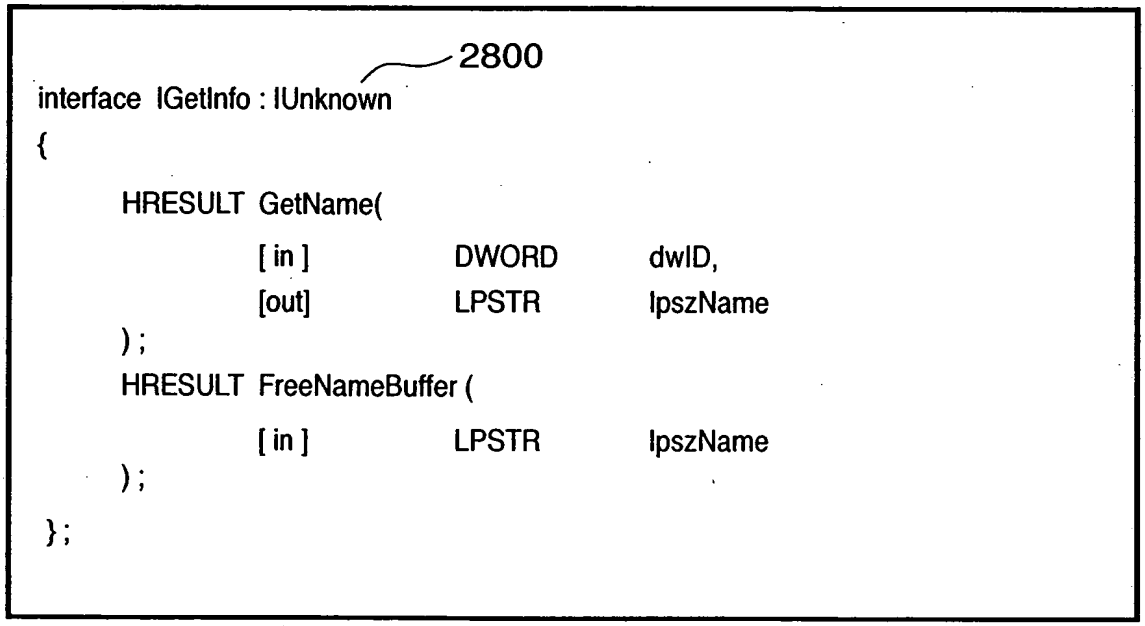

2800
interface IGetInfo : IUnknown
{
HRESULT GetName(
[in] DWORD dwID,
[out] LPSTR lpszName
);
HRESULT FreeNameBuffer (
[in] LPSTR lpszName
);
};

FIG. 29

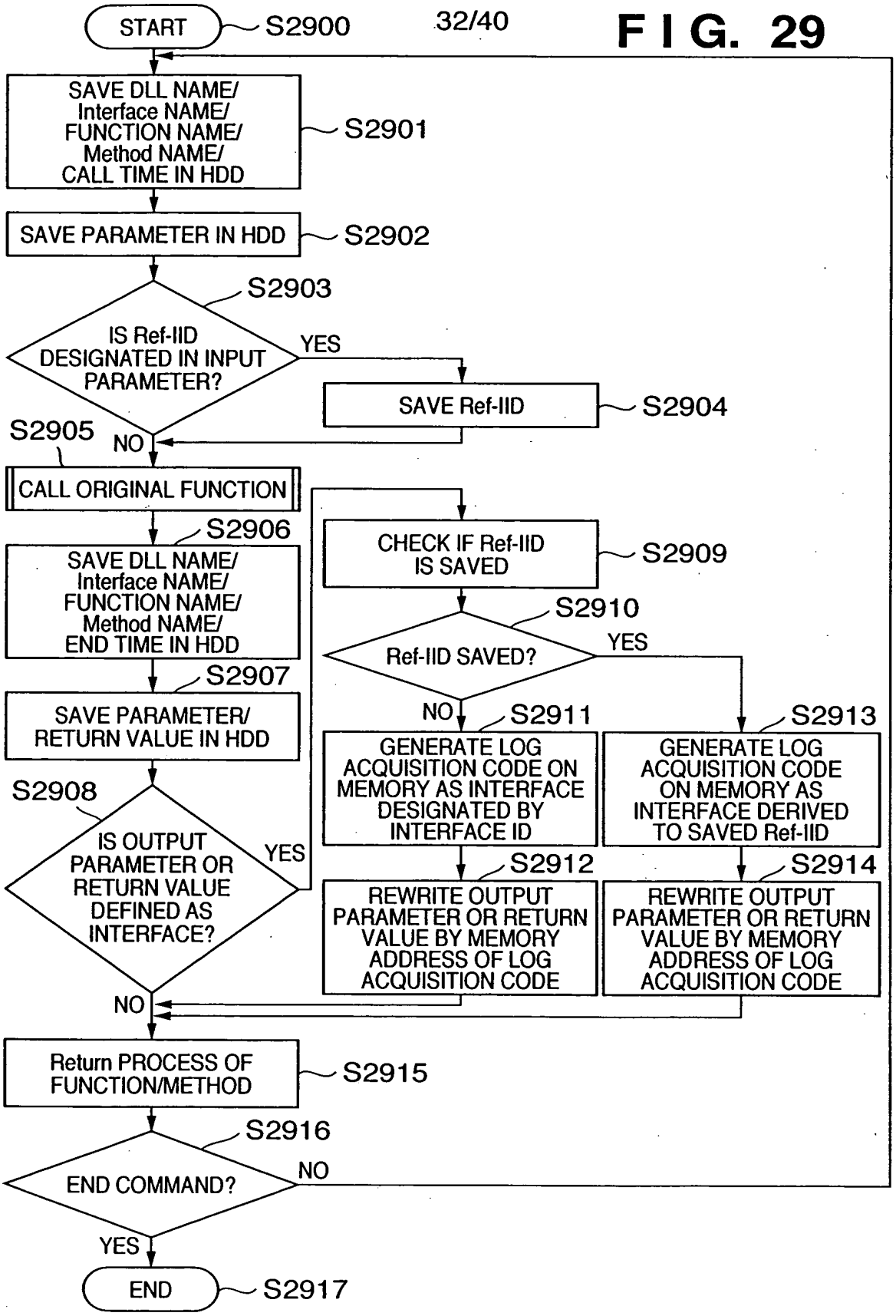

START
S2900
SAVE DLL NAME/ Interface NAME/ FUNCTION NAME/ Method NAME/ CALL TIME IN HDD
S2901
SAVE PARAMETER IN HDD
S2902
S2903
IS Ref-IID DESIGNATED IN INPUT PARAMETER?
YES
SAVE Ref-IID
S2904
S2905
NO
CALL ORIGINAL FUNCTION
S2906
SAVE DLL NAME/ Interface NAME/ FUNCTION NAME/ Method NAME/ END TIME IN HDD
S2907
SAVE PARAMETER/ RETURN VALUE IN HDD
S2908
IS OUTPUT PARAMETER OR RETURN VALUE DEFINED AS INTERFACE?
YES
NO
CHECK IF Ref-IID IS SAVED
S2909
S2910
Ref-IID SAVED?
YES
NO
S2911
GENERATE LOG ACQUISITION CODE ON MEMORY AS INTERFACE DESIGNATED BY INTERFACE ID
S2912
REWRITE OUTPUT PARAMETER OR RETURN VALUE BY MEMORY ADDRESS OF LOG ACQUISITION CODE
S2913
GENERATE LOG ACQUISITION CODE ON MEMORY AS INTERFACE DERIVED TO SAVED Ref-IID
S2914
REWRITE OUTPUT PARAMETER OR RETURN VALUE BY MEMORY ADDRESS OF LOG ACQUISITION CODE
Return PROCESS OF FUNCTION/METHOD
S2915
S2916
END COMMAND?
NO
YES
END
S2917

FIG. 30

```
MODULE NAME          GetInfo. dll
FUNCTION NAME :      DllGetClassObject                          3000
In ARGUMENT :        GUID* rclsid :   {abce80d7-9f46-11d1-882a-00c04fb961ec}
                     GUID* riid :     {00000001-0000-0000-c000-0000000046}
Out ARGUMENT :       void**ppv :      0x007a61c0/0x730e13e8(1930302440)
RETURN VALUE :       HRESULT :        0x00000000(0)                    3001
In TIME :            2002/03/25 22 : 24 : 12. 025          3002
Out TIME :           2002/03/25 22 : 24 : 12. 035

Interface NAME :     GetInfo
Method NAME :        GetName
In ARGUMENT :        DWORD dwID :     1
                     LPSTR lpszName   "Name-1"
RETURN VALUE :       HRESULT :        0x00000000(0)
In TIME :            2002/03/25 22 : 24 : 12. 046
Out TIME :           2002/03/25 22 : 24 : 12. 057

Interface NAME :     GetInfo
Method NAME :        FreeNameBuffer
In ARGUMENT :        LPSTR lpszName   "Name-1"
RETURN VALUE :       HRESULT :        0x00000000(0)
In TIME :            2002/03/25 22 : 24 : 12. 068
Out TIME :           2002/03/25 22 : 24 : 12. 079


Interface NAME :     GetInfo
Method NAME :        GetName
In ARGUMENT :        DWORD dwID :     2
                     LPSTR lpszName   "Name-2"
RETURN VALUE :       HRESULT :        0x00000000(0)
In TIME :            2002/03/25 22 : 24 : 12. 089
Out TIME :           2002/03/25 22 : 24 : 13. 000

Interface NAME :     GetInfo
Method NAME :        FreeNameBuffer
In ARGUMENT :        LPSTR lpszName   "Name-2"
RETURN VALUE :       HRESULT :        0x00000000(0)
In TIME :            2002/03/25 22 : 24 : 13. 011
Out TIME :           2002/03/25 22 : 24 : 13. 022

...
```

FIG. 31

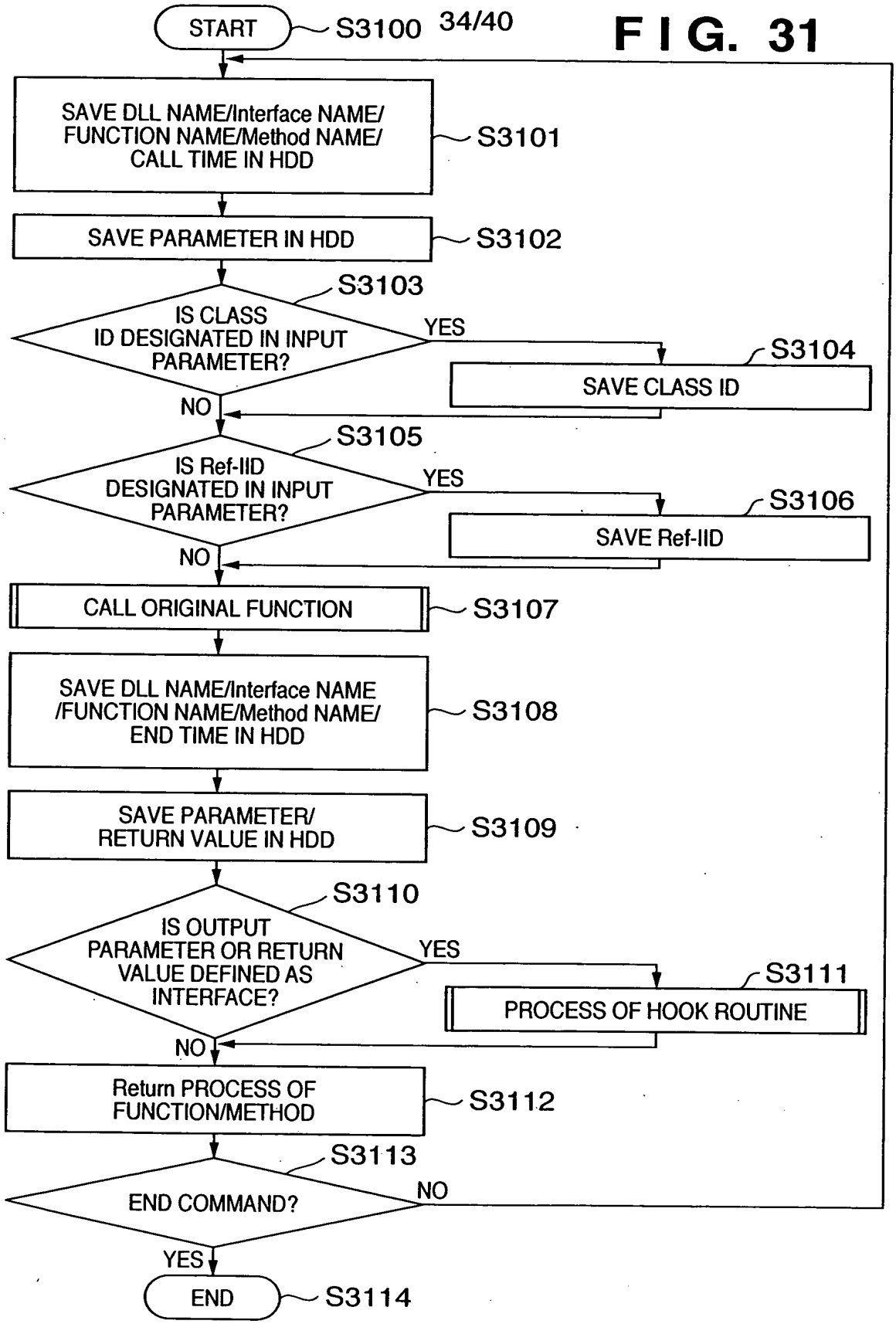

START
S3100
SAVE DLL NAME/Interface NAME/ FUNCTION NAME/Method NAME/ CALL TIME IN HDD
S3101
SAVE PARAMETER IN HDD
S3102
S3103
IS CLASS ID DESIGNATED IN INPUT PARAMETER?
YES
S3104
SAVE CLASS ID
NO
S3105
IS Ref-IID DESIGNATED IN INPUT PARAMETER?
YES
S3106
SAVE Ref-IID
NO
CALL ORIGINAL FUNCTION
S3107
SAVE DLL NAME/Interface NAME /FUNCTION NAME/Method NAME/ END TIME IN HDD
S3108
SAVE PARAMETER/ RETURN VALUE IN HDD
S3109
S3110
IS OUTPUT PARAMETER OR RETURN VALUE DEFINED AS INTERFACE?
YES
S3111
PROCESS OF HOOK ROUTINE
NO
Return PROCESS OF FUNCTION/METHOD
S3112
S3113
END COMMAND?
NO
YES
END
S3114

FIG. 32

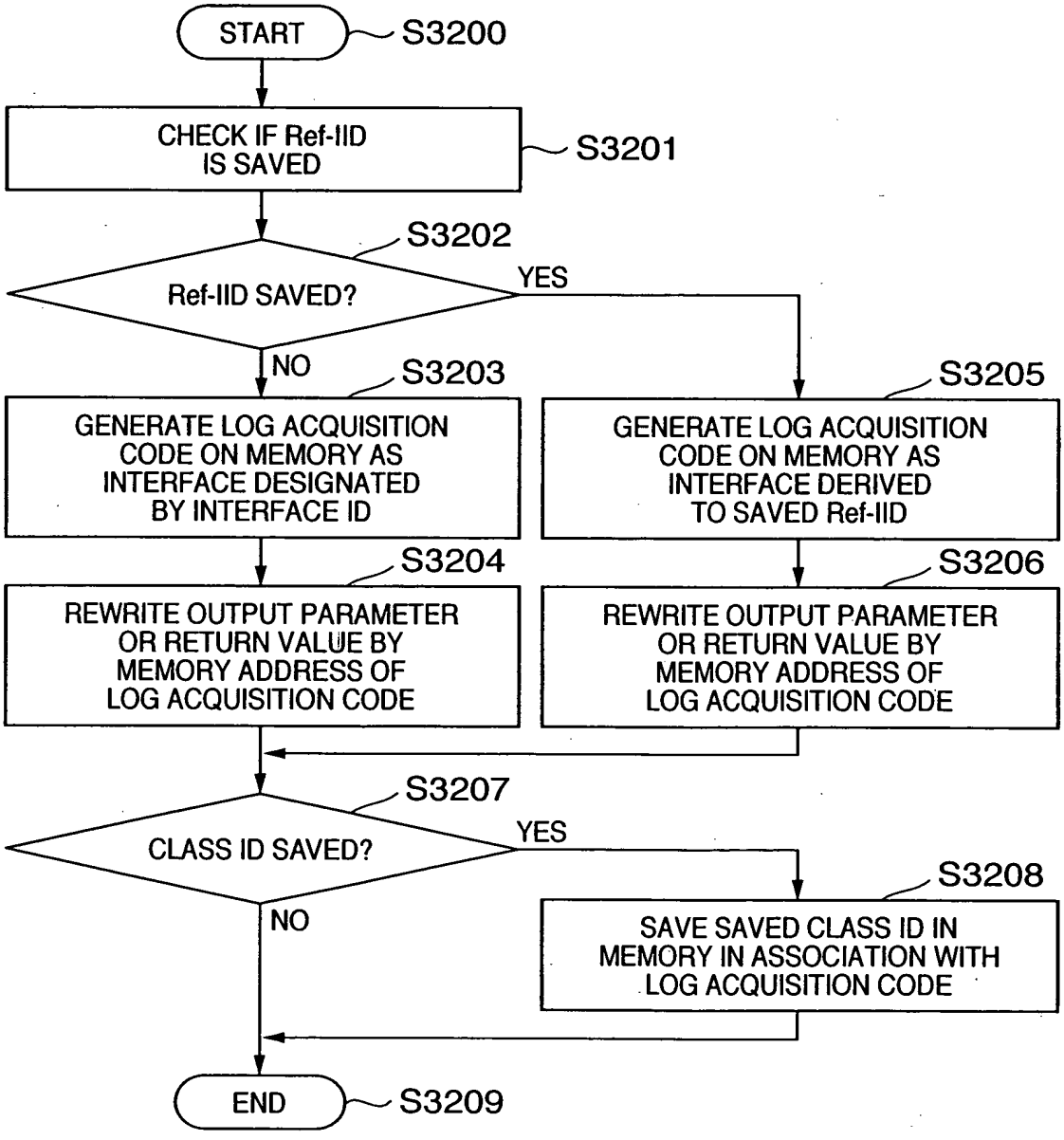

START
S3200
CHECK IF Ref-IID IS SAVED
S3201
S3202
Ref-IID SAVED?
YES
NO
S3203
GENERATE LOG ACQUISITION CODE ON MEMORY AS INTERFACE DESIGNATED BY INTERFACE ID
S3204
REWRITE OUTPUT PARAMETER OR RETURN VALUE BY MEMORY ADDRESS OF LOG ACQUISITION CODE
S3205
GENERATE LOG ACQUISITION CODE ON MEMORY AS INTERFACE DERIVED TO SAVED Ref-IID
S3206
REWRITE OUTPUT PARAMETER OR RETURN VALUE BY MEMORY ADDRESS OF LOG ACQUISITION CODE
S3207
CLASS ID SAVED?
YES
NO
S3208
SAVE SAVED CLASS ID IN MEMORY IN ASSOCIATION WITH LOG ACQUISITION CODE
END
S3209

FIG. 33

MODULE NAME	GetInfo. dll
FUNCTION NAME :	DllGetClassObject
In ARGUMENT :	GUID* rclsid : {abce80d7-9f46-11d1-882a-00c04fb961ec}
	GUID* riid : {00000001-0000-0000-c000-0000000046}
Out ARGUMENT :	void**ppv : 0x007a61c0/0x730e13e8(1930302440)
RETURN VALUE :	HRESULT : 0x00000000(0)
In TIME :	2002/03/25 22 : 24 : 12. 025
Out TIME :	2002/03/25 22 : 24 : 12. 035
MODULE NAME	GetInfo. dll
Interface NAME :	GetInfo
Method NAME :	GetName
In ARGUMENT :	DWORD dwID : 1
	LPSTR lpszName "Name-1"
RETURN VALUE :	HRESULT : 0x00000000(0)
In TIME :	2002/03/25 22 : 24 : 12. 046
Out TIME :	2002/03/25 22 : 24 : 12. 057
MODULE NAME	GetInfo. dll
Interface NAME :	GetInfo
Method NAME :	FreeNameBuffer
In ARGUMENT :	LPSTR lpszName "Name-1"
RETURN VALUE :	HRESULT : 0x00000000(0)
In TIME :	2002/03/25 22 : 24 : 12. 068
Out TIME :	2002/03/25 22 : 24 : 12. 079
MODULE NAME	GetInfo. dll
Interface NAME :	GetInfo
Method NAME :	GetName
In ARGUMENT :	DWORD dwID : 2
	LPSTR lpszName "Name-2"
RETURN VALUE :	HRESULT : 0x00000000(0)
In TIME :	2002/03/25 22 : 24 : 12. 089
Out TIME :	2002/03/25 22 : 24 : 13. 000
MODULE NAME	GetInfo. dll
Interface NAME :	GetInfo
Method NAME :	FreeNameBuffer
In ARGUMENT :	LPSTR lpszName "Name-2"
RETURN VALUE :	HRESULT : 0x00000000(0)
In TIME :	2002/03/25 22 : 24 : 13. 011
Out TIME :	2002/03/25 22 : 24 : 13. 022

...

3300
3301
3302
3303
3304
3305
3306

FIG. 34

```
[
  uuid(58DB5633-0694-4340-97CE-4E1AC6BFFBA7),    //TestDllStd.
  helpstring("TestDllStd Type Library For PAT"),
  version(1.0)
]

library TestDllStd
{
        typedef [public] struct
        {
                char chParam;
                unsigned char uchParam;
                short sParam;
                unsigned short usParam;
                int nParam;
                unsigned int unParam;
                long lParam;
                unsigned long ulParam;
                double dbParam;
                float fParam;
        } TESTSTRUCT;
        typedef [public] TESTSTRUCT *LPTESTSTRUCT;

//DEFINE_GUID(GUID_PROGID, 0x8e037d65, 0xefa0, 0x40e7, 0x91, 0x43, 0xef, 0x70, 0x56, 0x94, 0
0x79);
 [
  uuid(8E037D65-EFAO-40e7-9143-EF7056945B79),
  helpstring("TestDllStd.dll for PAT object." ),
 ]
        interface
        test
        {
                char _stdcall FuncCharStd([in] char chParam);
                char* _stdcall FuncPCharStd([in, out] char* lpchParam);

                TESTSTRUCT _stdcall FuncStructStd([in]TESTSTRUCT TestStruct);
                LPTESTSTRUCT _stdcall FuncPStructStd([in, out]LPTESTSTRUCT lpTestStruct);
        };
}
```

FIG. 35

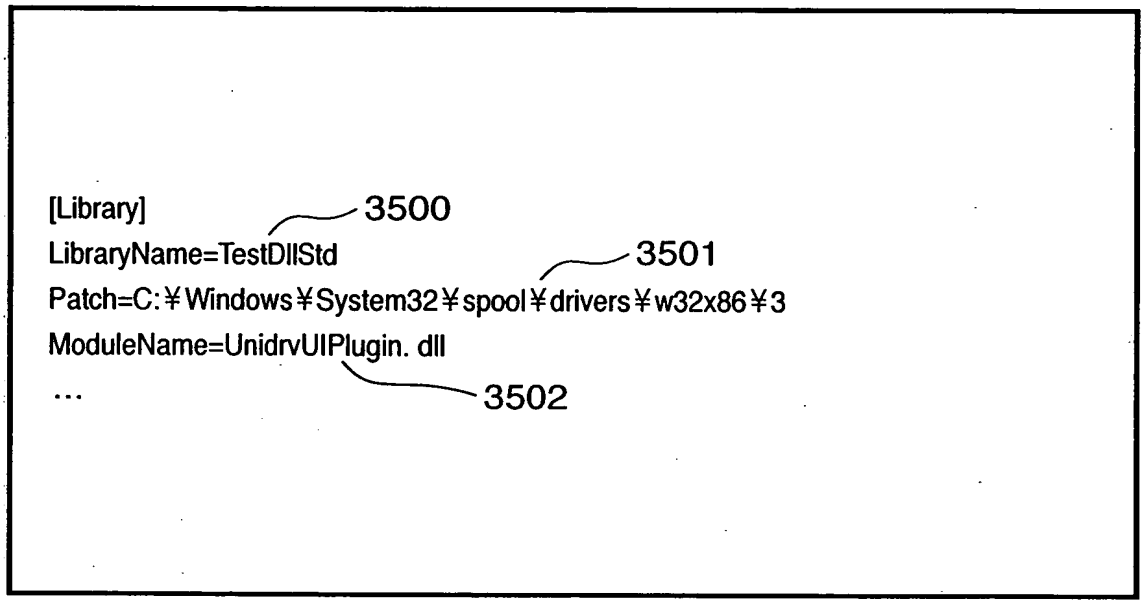

[Library]
3500
LibraryName=TestDllStd
3501
Patch=C:¥Windows¥System32¥spool¥drivers¥w32x86¥3
ModuleName=UnidrvUIPlugin. dll
...
3502

FIG. 36

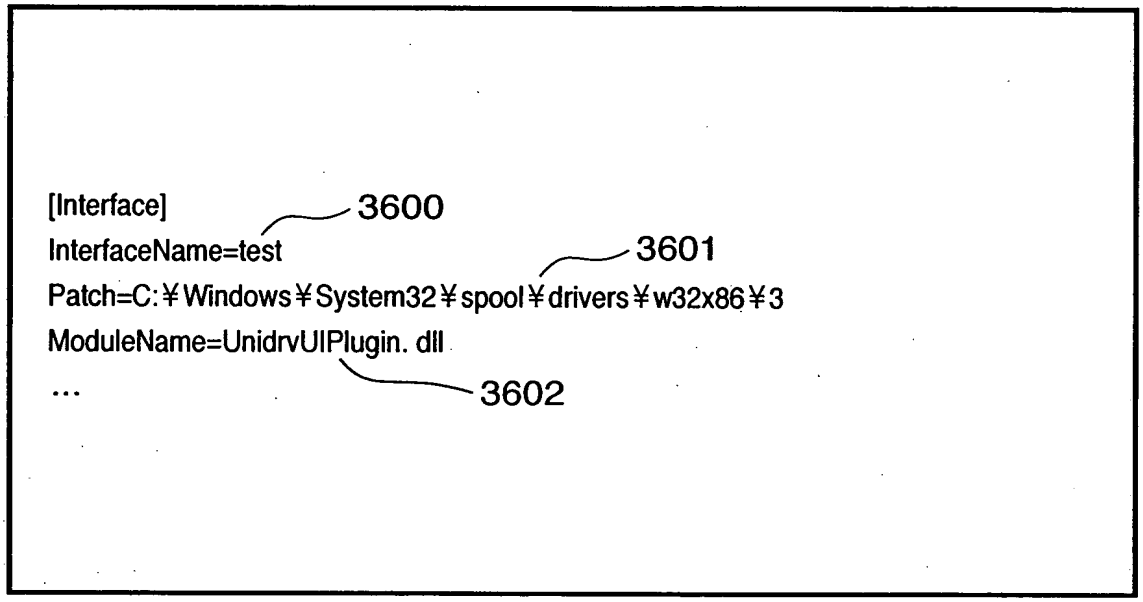
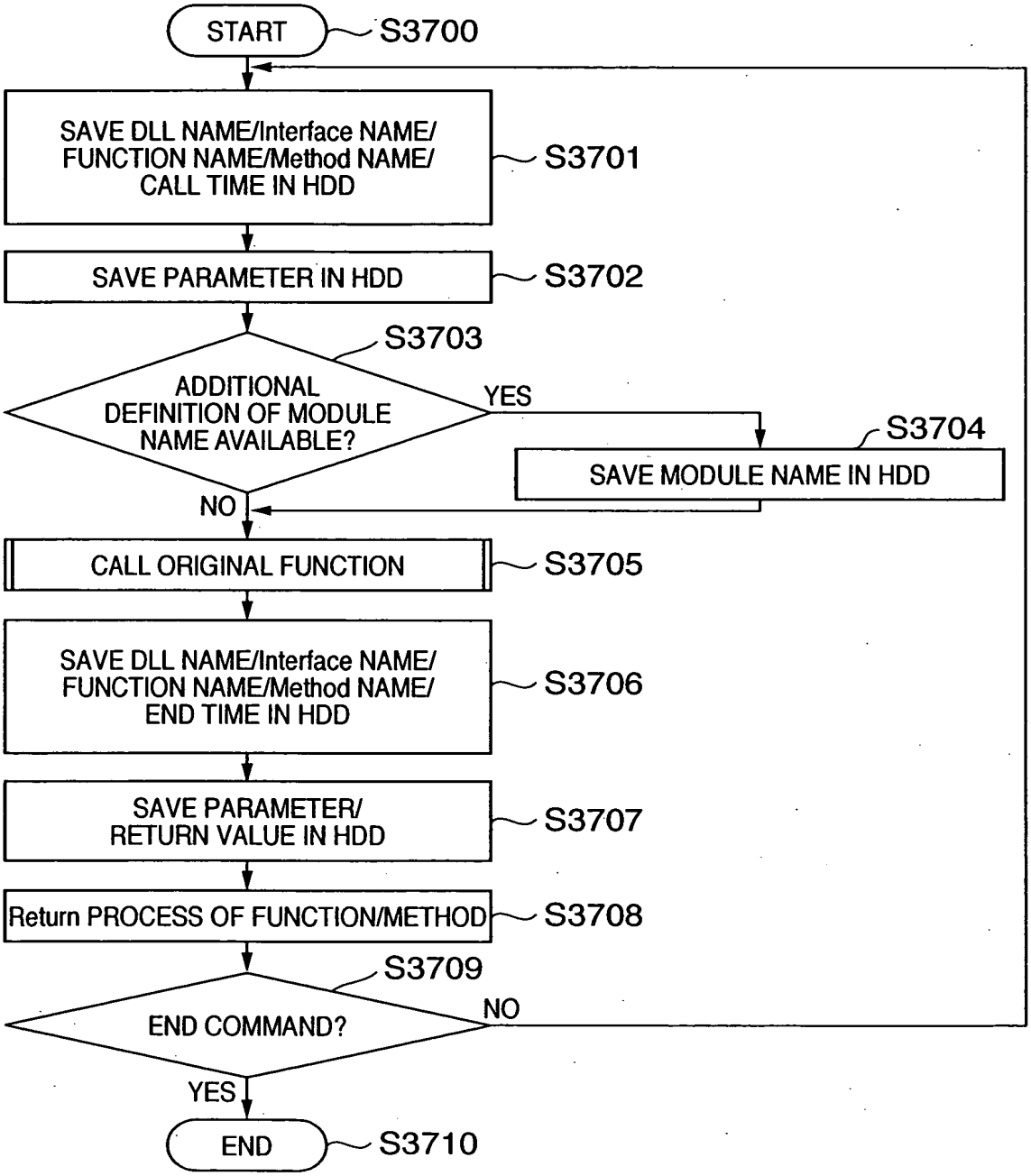

[Interface]
3600
InterfaceName=test
3601
Patch=C:¥Windows¥System32¥spool¥drivers¥w32x86¥3
ModuleName=UnidrvUIPlugin. dll
...
3602

FIG. 37


START
S3700
SAVE DLL NAME/Interface NAME/ FUNCTION NAME/Method NAME/ CALL TIME IN HDD
S3701
SAVE PARAMETER IN HDD
S3702
S3703
ADDITIONAL DEFINITION OF MODULE NAME AVAILABLE?
YES
S3704
SAVE MODULE NAME IN HDD
NO
CALL ORIGINAL FUNCTION
S3705
SAVE DLL NAME/Interface NAME/ FUNCTION NAME/Method NAME/ END TIME IN HDD
S3706
SAVE PARAMETER/ RETURN VALUE IN HDD
S3707
Return PROCESS OF FUNCTION/METHOD
S3708
S3709
END COMMAND?
NO
YES
END
S3710